دانشکده‌ی مهندسی

گروه مهندسی کامپیوتر

پایان‌نامه کارشناسی سخت‌افزار کامپیوتر

# طراحی و پیاده‌سازی روبات سکوی حرکتی دو حسگره

یحیی حسن زاده نظرآبادی

سید میلاد محمدی

استاد راهنما:

دکتر احد هراتی

استاد داور:

دکتر عابدین واحدیان مظلوم

سه‌شنبه ۱۲ شهریور ۱۳۹۲

# به نام پروردگار یکتای بی‌همتای

# پیشگفتار ۱

برای زیستن دو قلب لازم است

قلبی که دوست بدارد، قلبی که دوستش بدارند

قلبی که هدیه کند، قلبی که بپذیرد

قلبی که بگوید، قلبی که جواب دهد

قلبی برای من، قلبی برای انسانی که می‌خواهم

تا انسان را کنار خود حس کنم......

.

شـاید یکی از سخت‌ترین بخش‌های نگارش این پروژه‌ی کارشناسی، همین پیشگفتار باشد! هر خط آن بیشتر از ده‌ها بار پاک شد و دوباره نوشـته شـد. ابتدا کلیشـه ای، سـپس خسته کننده، بعد اندکی ادبی و همین روال ادامه پیدا کرد. در حین نوشتن این پیشگفتار چندین کتاب علمی را ورق زدم و تازه فهمیدم قسمتی به نام پیشگفتار نیز وجود دارد! حداقل از زمانی که وارد دانشگاه شده‌ام یادم نمی‌آید پیش گفتار کتابی را خوانده باشـم! چه بد! یادم از دوران دبیرسـتان افتاد. آنجایی که کتاب های تسـت اندیشـه‌سازان را می خریدم. در آن زمان یکی از تفریحاتم خواندن مقدمه‌های زیبای کتاب های تسـت اندیشـه‌سـازان بود. مقدمه‌هایی که در مورد همه چیز بود به جز خود کتاب! این قضیه بسیار بسیار برایم جالب بود. شـعر بالا هم از یکی از این مقدمه ها هست. دکتر فرهاد میثمی، مدیر مسئول انتشارات اندیشه‌سازان، در یکی از مقدمه هایش نوشته‌بود که مقدمه (البته در مورد این پایان‌نامه می شود پیشگفتار!) جایی است که در آن نویسنده فارغ از تمامی چارچوب‌های از پیش تعیین شده، می‌نشیند و دو کلمه حرف صمیمانه با خواننده می زند! و از آنجا بود که من هم تصمیم گرفتم، این پیشگفتار را به سبک دیگری بنویسم. امیدوارم مورد پسند واقع شود.

از سـال ۱۳۸۶ که در رشـته‌ی مهندسـی برق دانشگاه فردوسی مشهد پذیرفته شدم، بسیار علاقه داشتم که در یک تیم روباتیک کار کنم. شـاید اصـلا به همین دلیل این رشـته را انتخاب کردم. از همان سـال بود که تیم روباتیکی درسـت کردم و دسـت و پا شـکسـته در زمینه‌ی روبات‌های مین یاب کار می‌کردم. اما برق از آنچه تصـورش را می‌کردم بسیار فاصله داشت. برای همین در سال ۱۳۸۸ از برق به کامپیوتر تغییر رشـته دادم. رشته‌ای که هنوز هم عاشقش هستم و با تمام وجود آنرا دوست دارم. آرام آرام کار روباتیک را توسعه دادیم و امروز آن تیم کوچک دست و پا شکسته دارای نام، شهرت، کادر فنی، آیین نامه، هدف بلندمدت و حتی آزمایشگاه خصوصی است. به پاس افتخار به تاریخ پر عظمت ایران، نام پارســـه (تخت جمشـــید) را بر آن نهادیم. کار کردن در این آزمایشـــگاه بر روی روباتهای مختلف، یکی از شـــیرین‌ترین و زیباترین بخش‌های زندگی من شـد. حتی ۴ بار هم مقام کسـب کردیم! دو مقام دوم کشـوری و دو مقام سـوم کشوری. خیلی شیرین، خیلی جالب! یک تعدادی هم مقاله از کارهایمان در کنفرانس ها و مجلات به چاپ رسـید. امیدواریم مورد قبول جامعه‌ی علمی کشـور باشـد. خلاصـه این تیم روباتیک شد همه چیز برای خیلی‌ها که به آن دلبسته شدند، از جمله خود من.

کارکردن بر روی این پایان نامه بیشـتر از ۴ سـال در همین تیم پارسه به طول انجامید. ابتدا قرار نبود پایان نامه باشد. قرار بود یک روبات باشـد برای شـرکت در مسـابقات روبوکاپ آزاد ایران که هر سـاله فروردین ماه برگزار می شـود. این رویداد به نوعی بزرگترین رویداد روباتیک خاورمیانه اسـت! روبات را سـاختیم و رفتیم. در آنجا شـگفتی سـاز مسابقات شدیم! و در کمال ناباوری به رتبه ی چهارم لیگ نمایش آزاد نایل شـدیم. البته رتبه ی چهارم به نوع خود زیاد جالب نیست! اما دو مسئله در آن نهفته بود: اول، اگر سوم می‌شدیم به ما جایزه می‌دادند و یعنی

فقط یک رتبه تا رفتن روی جایگاه و نور و فلش دوربین‌ها و مصاحبه و لوح و کاپ فاصله داشتیم! دوم، کمتر تیمی در اولین بار شرکت در این مسابقات رتبه‌ی خوب کسب می کند و ما میان حدود ۱۱ تیم شرکت کننده چهارم شده بودیم. همین انگیزه‌ای شد برای ادامه دادن تا آخر! گرچه کادر تیم تقریبا هر ۶ ماه یک بار کاملا عوض می‌شدند، خیلی‌ها می‌آمدند، خیلی‌ها می‌رفتند، خیلی‌ها اهداف دیگری داشتند. خیلی وقت‌ها کار آن طور که باید پیش نمی رفت و هزاران مشکل دیگر. اما مصمم بودیم که بسازیم و بسازیم و بسازیم. حالا کامل شد! فروردین ۱۳۹۲ برای اولین بار یک نتیجه‌ی درست و قابل گزارش از این پروژه گرفتیم و شد آنچه شد! بسیار خوشحال بودیم از اینکه بعد از ۴ سال زحمات ما به اصطلاح به نتیجه رسیدند. هنوز هم دست از کار نخواهیم کشید. در حال حاضر که نگارش این پایان نامه دارد به خط پایان خود نزدیک و نزدیک‌تر می شود، در حال طراحی نسل دوم این پروژه و بهینه سازی آن هستیم.

وظیفه‌ی خود می دانم که از زحمات تمامی اساتید عزیز و محترم در گروه کامپیوتر دانشگاه فردوسی مشهد تشکر کنم، همیشه شاگرد کوچک این بزرگواران و مدیون زحماتشان خواهم بود. تشکر ویژه‌ای دارم از استاد راهنمای این پایان‌نامه، جناب آقای دکتر هراتی که همواره مشوق ما بودند و با راهنمایی‌های سازنده روشن ساز راه ما در این پروژه شدند. همچنین از استاد بسیار گرامی، جناب آقای دکتر واحدیان که این افتخار را به ما دادند و زحمت داوری این پایان‌نامه را قبول فرمودند، سپاسگزاری ویژه‌ای دارم، همواره شاگرد این استاد بزرگ خواهم بود. از تمامی نسل‌های تیم پارسه از سال ۱۳۸۶ تا کنون تشکر می‌کنم. حالا که فکر می کنم، اگر بخواهم تک تک اسامی آنها را بنویسم شاید ده‌ها خط بشود! از همه‌ی آنها چیزهای فراوانی یاد گرفتم. بدون شک بدون وجود بسیاری از آنها، حتی آنهایی که در سایر پروژه های روباتیک کار می‌کردند، اتمام این پروژه غیر ممکن بود. به خصوص از دوستان عزیزم در بخش دانش‌آموزی تیم پارسه سجاد غریب نواز و امین علیزاده، که در پیشرفت عملیاتی این پروژه بسیار زحمت کشیدند، کمال تشکر را دارم. از بهترین دوست دوران دانشکده، میلاد محمدی عزیز، نویسنده‌ی دیگر این پایان‌نامه، به خاطر تمامی زحماتی که در پیشبرد این پروژه کشید و تمام چیزهایی که به من یاد، سپاس گزارم.

در پایان، این پایان‌نامه را به با ارزش‌ترین افراد زندگی خود، پدر بزرگوار، مادر مهربان و برادر عزیزم تقدیم می‌کنم و برای روح بزرگوار مادربزگ گرامی‌ام به پاس تمامی زحمات دلسوزانه ای که از دوران کودکی تا جوانی برایم متحمل شد، از درگاه پروردگار بزرگ، آرامش و رستگاری می‌طلبم. جا دارد تا از دایی عزیزم، جناب آقای احمد ابوترابی، کسی که در سن کودکی برای اولین بار مرا با دنیای الکترونیک آشنا کرد و بدون وجود او قطعا امروز در این رشته ی تحصیلی قدم برنمی داشتم، کمال تشکر را داشته باشم.

زندگی صحنه ی یکتای هنرمندی ماست....

هرکسی نغمه ی خود خواند و از صحنه رود....

صحنه پیوسته به جاست.....

خرم آن نغمه که مردم بسپارند به یاد.....

یحیی حسن زاده نظرآبادی

۳ شهریور ماه ۱۳۹۲ خورشیدی

مشهد

# پیشگفتار ۲

همان‌طور که امروزه در تمام رسانه‌ها و اطراف خود شاهد هستیم علم رباتیک و اتوماسیون در حال گسترش و به عبارت بهتر تسخیر تمام زمینه‌های کاری است، چیزی که عده‌ای از بابت آن ابراز نگرانی می‌کنند و علت این نگرانی را نیز این‌طور بیان می‌کنند که با این گسترش روزافزون فرصتهای شغلی برای انسانها پایین می‌آید، این در حالی است که اخیرا یکی از مشهورترین سایت‌های خبررسانی در آمریکا اعلام کرد طبق پیش‌بینی محققین جامعه شناس تا سال ۲۰۲۰ حدود ۱۶ میلیون فرصت شغلی مرتبط با روباتیک در آمریکا ایجاد خواهد شد که خود رقم بسیار بالا و خیره کننده‌ای در زمینه‌ی ایجاد شغل می‌باشد. به نظر بنده مهندسی یک هنر است و شاهکار تمام دوران این هنر چیزی نیست جز رباتیک!

این پایان‌نامه حاصل چندین سال تجربه‌ی من و دوستانم در زمینه‌ی رباتیک است و جا دارد همین جا از تمام کسانی که ما را در این مسیر یاری کردند تشکر ویژه و صمیمانه‌ی خود را اعلام دارم و بخصوص تشکر ویژه از جناب آقای دکتر احد هراتی که در پست استاد راهنمایی قبول زحمت فرمودند و همچنین دکتر عابدین واحدیان مظلوم که ایشان نیز به عنوان استاد داور افتخار دادند و ما را از محبت‌های بی‌دریغ خود بهره‌مند ساختند.

در پایان امیدوارم با ارائه‌ی این پایان نامه توانسته باشیم کمکی اگرچه اندک به پیشرفت علمی و عملی این مرز و بوم کرده باشیم.

سید میلاد محمدی

۱۳۹۲/۶/۷

**فهرست مطالب**

# چکیده

روبات‌های هوشمند دارای حسگر، اغلب نمی توانند به حداکثر دقت لازم در اجرای الگوریتم‌های خود دست یابند. این موضوع ناشی از عوامل مختلفی است که سبب می شود روبات نتواند تصویر و درک درستی از محیط اطراف خود فراهم آورد. در این پروژه ی کارشناسی ابتدا این عوامل بررسی می شود. سپس برای رفع این مشکل یک روبات سکو پیشنهاد می شود. این روبات سکو DSPR نام دارد. روبات "Dual Sensor Platform" با به اختصار DSPR یک Platform کاملا هوشمند و حاصل تلاش دو نسل در طی ۴ سال پیرامون مطالعات، شبیه‌سازی و محاسبات آزمایشگاه رباتیک پارسه می باشد. این روبات با هدف کمک به برطرف کردن مشکل جمع‌آوری اطلاعات که امروزه اغلب روبات‌های هوشمند دارای حسگر با آن مواجه هستند، طراحی شده است. DSPR در مقایسه با روبات‌های مشابه خود تجسمی کامل‌تر و دقیق‌تر از محیط اطراف را در اختیار پردازنده‌ی الگوریتم خود قرار می‌دهد. DSPR قادر است تنها با داشتن دو عدد حسگر فراصوت تمامی موانع موجود در ۳۶۰ درجه‌ی اطراف را تا شعاع معین و قابل تنظیم تشخیص داده، بدون برخورد با موانع در میان آنها حرکت کند، با استفاده از Encoder که روی موتورهای آن وجود دارد مختصات روبات را گزارش بدهد و توسط یک ماژول فرستنده-گیرنده بی سیم با یک رایانه ارتباط برقرار کرده و تبادل اطلاعات نماید. این روبات توانایی اجرای الگوریتم‌هایی مثل جستجو، جاروب کردن، نقشه برداری و مشابه این موارد را دارد. DSPR می تواند به عنوان یک Platform برای اجرای الگوریتمهای مختلف بدون کمترین تغییرات نرم افراری استفاده شود. این روبات می تواند جایگزین مناسبی برای سایر روباتهای موجود مانند مین یاب، جستجوگر، میکرو موس و مانند آنها باشد. DSPR یک Platform بومی است. به این معنی که متناسب با نیازهای جامعه روباتیک داخل کشور طراحی شده است و تمامی قطعات و فناوری‌های به کار رفته در آن در بازار بومی ایران قابل تهیه می‌باشد.

# کلمات کلیدی

روبات، موتور DC، موتور پله‌ای، موتور سرو، گذرگاه ورودی خروجی، نویز، حسگر، میکروکنتلر AVR، زبان برنامه نویسی ++C، ارتباط I2C، ارتباط سریال

# فصل اول

# مقدمه

## ۱-۱-مفاهیم

Platform در لغت به معنای سـکو می‌باشد. در اصطلاح علم روباتیک Platfrom یک روبات تک منظوره، چند منظوره یا همه منظوره اسـت که به طور خاص یا به صورت کلی قابل برنامه نویسی می‌باشد.[۱] بدیهی است روباتی با چنین شرایط باید تا حد امکان استانداردسازی شـده باشـد و دسترسی های کاربر یا همان برنامه‌ریز روبات را از سطحی به بعد استاندارد یا محدود نماید. محدود کردن دسترسی یک کاربر از یک سطح خاص به بعد به معنای فراهم آوردن تمامی واسط‌های مورد نیاز برای آن کاربر و قطع ارتباط مستقیم کاربر با سطوح پایین‌تر از سطح مورد نظر اسـت. برای مثال یک روبات Platform که از این به بعد در این پروژه **سکو** نامیده می‌شـود، می تواند دسـترسـی سخت افزار خود را محدود نماید. در این صـورت کاربر توانایی ارتباط مستقیم با سخت افزار را نخواهد داشت. اگر در این شرایط کاربر بخواهد حرکت جلو رفتن را به روبات دسـتور بدهد, باید این عمل را از طریق یک واسط که می تواند مثلا یک تابع به نام up() باشد فراخوانی نماید و نمیتواند مستقیما به تک‌تک موتورهای روبات فرمان بدهد. اسـتاندارد کردن دسـترسی کاربر به روبات تقریبا شبیه محدود کردن سطح دسترسی است، با این تفاوت که در این حالت ارتباط مستقیم کاربر با سطوح پایین‌تر قطع نخواهد شد و کاربر می تواند به طور مستقیم با سطوح پایین‌تر در تماس باشد. در مثال قبل اسـتاندارد کردن سـطح دسـترسـی کاربر به روبات به این معنی خواهد بود که کاربر علاوه بر امکان اسـتفاده از تابع up() می تواند مسـتقیما با دستورات سطح پایین تر، کنترل روبات را در دست بگیرد. در این حالت قطعا قدرت عمل کاربر بیشتر می شود و می تواند دامنه‌ی کنترل خود را بر روی روبات افزایش دهد. برای مثال در این حالت کاربر ممکن است بخواهد سرعت تک تک چرخ‌ها را تنظیم کند. در حالی که در وضـعیتی که دسترسی کاربر بر روی سخت افزار روبات محدود شده بود، شاید امکان کنترل سرعت چرخ‌ها از طرف طراح و سازنده‌ی روبات پیش‌بینی نشده باشد.

اسـتانداردسـازی یک روبات سـکو به معنی تطبیق هرچه بیشتر آن با شرایط مختلف می‌باشد. روباتی که قرار است به عنوان یک سکو طراحی شود باید تا حد امکان دامنه‌ی وسیعی از سازگاری با شرایط مختلف را در بر بگیرد. به عنوان یک مثال بسیار ساده، بدیهی است روباتی که به عنوان یک Platform قرار اسـت استفاده شود باید تا حد امکان تغذیه‌ی خود را حمل کند. چنین روباتی نباید سیستم تغذیه‌ای در خارج از خود داشـته باشـد. زیرا غالبا شـرایط محیطی که روبات سـکو قرار است در آن استفاده شود از قبل مشخص نیست. ممکن است یک سکوی سـاده برای نقشه‌برداری، مکان‌یابی، عملیات رفت و برگشت مکرر و موارد مشابه استفاده شود. بدیهی است که شرایط هر یک از این محیط‌ها با وجود شباهت‌های کلی، تفاوت‌های بنیادین دارد.

## ۱-۲ معرفی چند روبات سکوی نمونه

### ۱-۲-۱ آسیمو (ASIMO)

تا به امروزه روبات‌های سـکو بسیاری ساخته شده‌اند. به طور قطعی نمی‌توان در مورد اولین یا قدیمی‌ترین روبات سکو اظهار نظر کرد. طی یک بررسـی انجام شـده، می توان ادعا کرد روبات آسـیمو (ASIMO) یکی از قدیمی‌ترین سکوهای شناخته شده در سطح جهان می‌باشد. آسیمو یک روبات انسان نما است که با هدف کمک کردن به انسان طراحی و ساخته شده است. اولین نسل این روبات با نام E0 در سال ۱۹۸۶ میلادی رونمایی شـد. تا به امروز ۱۰ نسـل از این روبات از طرف شـرکت سـازنده ی آن هوندا (HONDA) به جهان معرفی شـده اند. در شکل شـماره ۱ آخرین نسل این روبات را که در سال ۲۰۰۰ رونمایی گردید و نام آسیمو بر آن نهاده شد، نشان داده شده است. از جمله ویژگی های آسـیمو می‌توان به سـبک وزن بودن آن، فناوری پیشرفته ی راه رفتن، پارامترهای عملیاتی عریض بازوها، ساده بودن به کارگیری آن و طراحی کاربرپسند آن اشاره کرد.[۲]



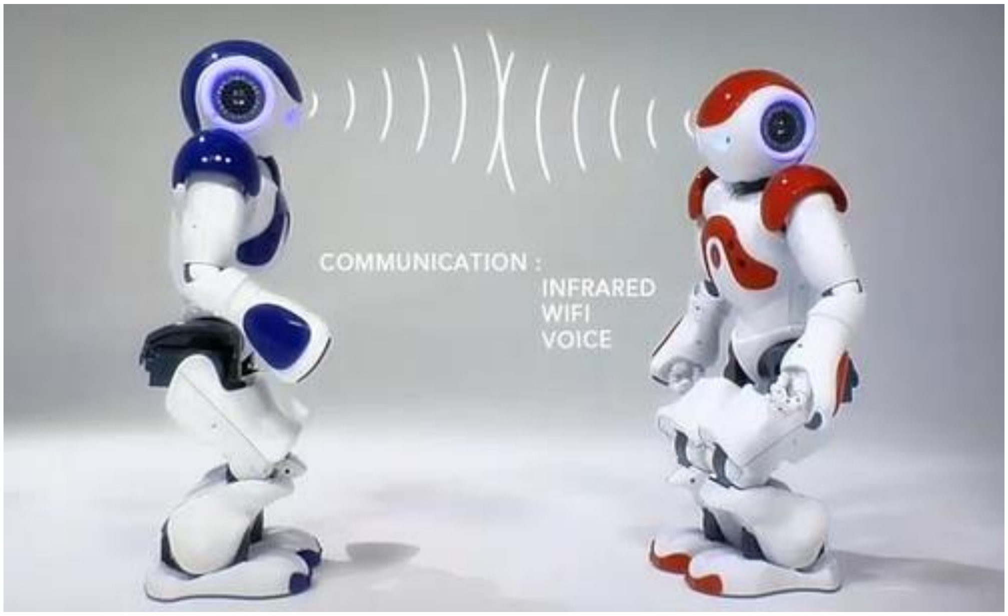

*شکل ۱- ۱ روبات آسیمو، یکی از معروفترین روباتهای انسان‌نمای جهان که به عنوان یک سکو استفاده می‌شود.*

### ۱-۲-۲ نائو (NAO)

روبات نائو (NAO) یک روبات انسان‌نمای کاملا هوشمند همه منظوره می‌باشد که در سال ۲۰۰۷ توسط کمپانی Aldebaran Robotics معرفی شد. این روبات از سال ۲۰۰۹ به طور رسمی به عنوان روبات استاندارد در لیگ Standard Platform در مسابقات روباتیک جهانی قرار گرفت. هدف از بکارگیری نائو در این لیگ، توسعه‌ی الگوریتم‌های تحقیقاتی و عملیاتی فوتبال می‌باشد. این روبات باید قادر باشد تا در غالب یک تیم با روبات‌های همتای خود به صورت کاملا هوشمند فوتبال بازی کند. بدون شک روبات نائو یکی از پیشرفته ترین سکوهای انسان‌نما می‌باشد. از جمله ویژگی‌های منحصر به فرد این روبات می‌توان به موارد زیر اشاره کرد:[۱, ۳]

- پردازش تصاویر به کمک دوربین
- تشخیص اشیاء
- تشخیص چهره و احراز هویت
- تشخیص صدا و صحبت
- قابلیت تبدیل متن به گفتار با پشتیبانی از ۹ زبان
- سیستم جلوگیری از سقوط
- سیستم جلوگیری از برخورد
- ارتباط با یکدیگر از طریق سیستم WiFi

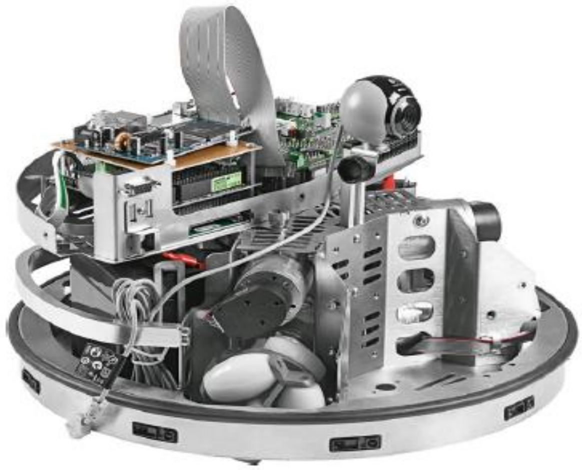
نائو امروزه بیشـتر یک روبات آموزشـی تحقیقاتی با دامنه‌ی بسـیار وسـیعی از کاربردهاست. از جمله ی کاربردهای این روبات می‌توان به کمک در تحقیقات بر روبات‌های انسان‌نما، مکان‌یابی، کاربردهای روباتهای روان‌شناس، هوش‌مصنوعی، همزمانی حرکت روبات و همچنین در بحث آموزش در موارد برنامه نویسـی، مسـائل فیزیک و ریاضـی روباتیک، مکاترونیک، هوشمندسازی، بازی‌سازی و نمایش استفاده کرد. در شکل شماره ۲ یک نمونه از روبات نائو به نام Y21 نشان داده شده است.[۳]




*شکل ۱-۲ روباتهای نائو به عنوان یکی از مشهورترین روباتهای سکوی انسان نمای تحقیقاتی قادر به ارتباط با یکدیگر از طریق صدا، مادون قرمز و شبکه ارتباط بی‌سیم می‌باشند*

### ۱-۲-۳ روبوتینو (Robotino)

برخلاف دو نمونه‌ی ذکر شـده‌ی قبلی (نائو و آسـیمو)، روبوتینو (Robotino) یک سـکوی انسـان نما نیسـت. مطابق شکل شماره ۱-۳ روبوتینو یک سـکوی سـخت افزاری است که از سال ۲۰۱۱ توسط شرکت Festo طراحی و ساخته و معرفی شده است. این روبات مرتبا توسط شرکت سازنده از نظر سخت افزاری و نرم افزاری به‌روزرسانی می شود. این روبات قادر است ماموریت‌های مشخصی را انجام دهد. مانند جابجایی اجسام به صورت دو بعدی، حرکت های ساده، مکان یابی، تشخیص موانع به کمک حسگرها و ارتباط بی سیم با روبات های نظیر خود.



شکل ۱ - ۳ روبوتینو یک سکوی سخت افزاری قابل برنامه‌ریزی می‌باشد.

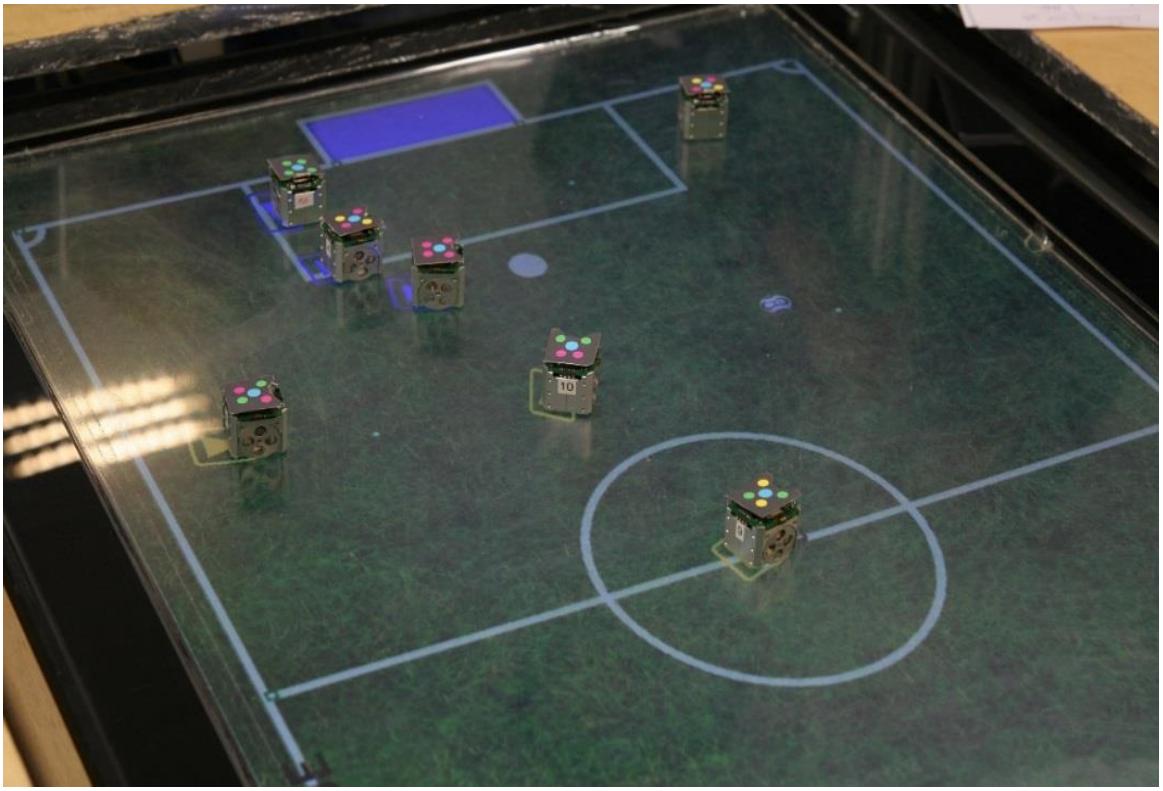
با وجود اینکه روبوتینو محبوبیت نائو در بین برنامه‌نویسان روباتیک و شهرت جهانی آسیمو را ندارد، اما امروزه در بین علاقه مندان به روباتیک حرفه‌ای جایگاه انکارناپذیری دارد. از سال ۲۰۱۲ با پشتیبانی شرکت Festo، یک لیگ کاملا مستقل به نام Logistics در مسابقات جهانی بوجود آمد. این لیگ دارای قوانین کاملا مشخص و مستقل، هدف، ماموریت و چشم انداز طولانی مدت می باشد. لیگ Logistics روبوتینو را به عنوان تنها روبات مجاز به استفاده در این لیگ معرفی کرده است. تمامی شرکت‌کنندگان در این لیگ موظف هستند حداقل ۳ عدد روبوتینو خریداری کرده و به برنامه نویسی آنها برای اجرای ماموریت مشخص بپردازند. در لیگ مسابقات جهانی روباتیک، هر تیم از ۳ ربوتینو تشکیل می‌شود که با همکاری یکدیگر باید چالش Logistic یک سیستم تولید ناشناخته را حل کنند. تیم ها حق دستکاری سخت‌افزاری روبات و تغییر در قابلیت‌های سخت افزاری آن را نخواهند داشت. آنها باید فقط به برنامه نویسی این روبات بپردازند. شاید یکی از نقاطی که مفهوم واژه ی Platform در علم روباتیک به وضوح احساس می شود، همین لیگ باشد. در شکل شماره ۱-۴ یک نمونه از محیط عملیاتی ربوتینو در مسابقات روباتیک جهانی نمایش داده شده است[۴].



شکل ۱ - ۴ محیط عملیاتی روبات های روبوتینو در مسابقات ربوکاپ آزاد آلمان ۲۰۱۳

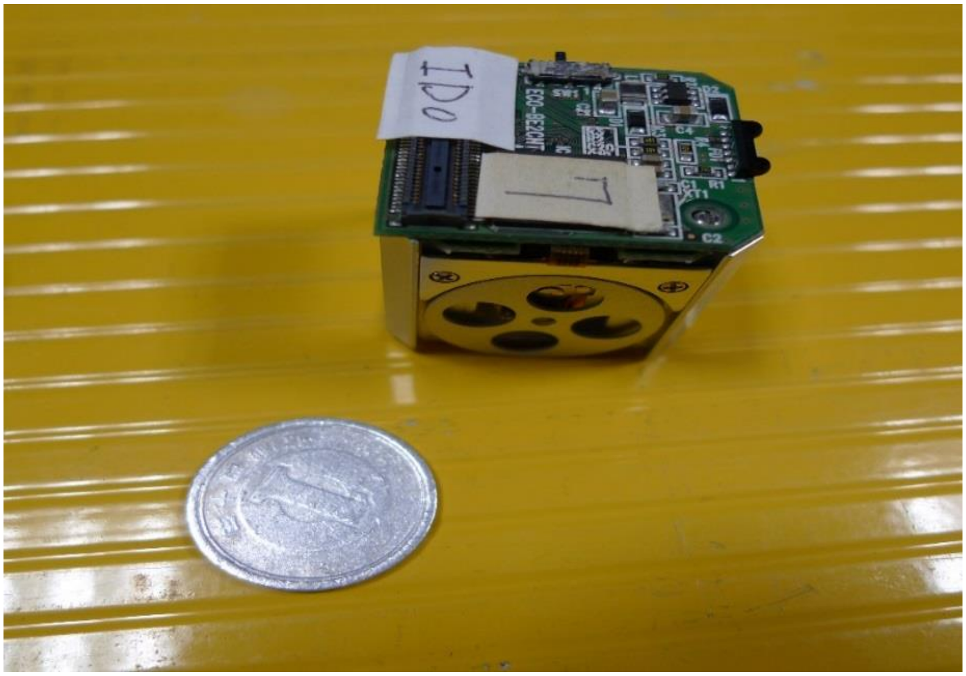
### ۱-۲-۴ روبات های واقعیت ترکیبی

واقعیت ترکیبی (Mixed Reality) یکی از لیگ‌های مسابقات جهانی روباتیک است. این لیگ برمبنای یک مسابقه‌ی فوتبال با همان قوانین و مقررات مرسوم فوتبال می‌باشد. شکل ۱-۵ یک نما از زمین این مسابقه را نشان می‌دهد. الگوریتم و برنامه ی هر یک از دو تیم فوتبال بر روی یک سرور اجرا شده و بازی روی یک صفحه ی مجازی نشان داده می‌شود. هم زمان روباتهای کوچک هر تیم، مطابق با یکی از بازیکنان مشخص صفحه‌ی مجازی حرکت کرده و نقش آن بازیکن را در دنیای واقعی اجرا می‌کنند. نامگذاری "واقعیت ترکیبی" به همین دلیل می‌باشد.



*شکل ۱ - ۵ زمین مسابقه‌ی روباتهای واقعیت ترکیبی، این روباتها یک سکوی استاندارد قابل برنامه‌ریزی هستند*

روبات‌های استفاده شده در این مسابقه، روباتهای مخصوصی بوده که به MR Platform مشهور هستند. این روباتها توسط شرکت های مختلفی تولید می‌شوند و در اختیار شرکت کنندگان در این لیگ قرار می‌گیرند. این روباتها همانطور که از نامشان پیداست یک سکوی سخت افزاری هستند و شرایط استانداردی را از جهت سخت افزاری برای مصرف کننده پدید می‌آورند. شرکت کنندگان در این لیگ باید این روبات‌ها را تهیه کرده و الگوریتم‌های خود را بر روی آنها برنامه‌نویسی کنند. شرکت کنندگان حق دستکاری روبات یا توابع سطح پایین آنرا ندارند و فقط می‌توانند برنامه نویسی خود را با استفاده از توابع سطح بالایی که در اختیار آنها گذاشته می شود، انجام دهند. شکل ۱-۶ یک نمونه از روبات‌های MR Platform را از نمای نزدیک نشان می دهد.[۵]



*شکل ۱- ۶ MR Platforms یک سکوی استاندار هوشمند بسیار کوچک قابل برنامه ریزی برای اجرای مسابقه‌ی فوتبال*

### ۱-۲-۵ ادی(Eddie)

ادی (Eddie) روباتی سـکو اسـت که توسـط شـرکت Parallax به سـفارش شرکت Microsoft ساخته شد. این روبات یک سکوی همه منظوره بوده و برای اغلب کاربردهای هوش مصـنوعی دارای کارایی و کارکرد مناسبی است. شکل ۱-۷ نمونه ی مشابه ساخته شده‌ی داخلی از این روبات را که توسـط یکی از مولفان این پایان‌نامه (م.محمدی) طراحی و ساخته شده است، نشان می‌دهد. این روبات دارای دو نوع حسگر و یک دوربین Kinect اسـت. حسـگرهای آن به ترتیب از نوع مادون قرمز و فراصـوت هستند. با توجه به قابلیتهای فراوانی که دوربین Kinect در اختیار کاربران خود قرار می‌دهد، این روبات نیز به قابلیت‌های منحصر به فردی به سبب داشتن این دوربین مجهز شده است.[۶, ۷]

دوربین Kinect توسـط شرکت Microsoft طراحی و ساخته شده است. این شرکت کتابخانه‌ها و امکانات نرم‌افزاری فراوانی را برای کاربران این نوع دوربین فراهم کرده اسـت. به نحوی که تا حد ممکن عملیات پردازشـی تصویر را برای کاربر خود به حداقل رساند. مثلا یکی از قابلیت‌های منحصـر به فرد این دوربین دادن ماتریس خروجی عمق در کنار ماتریس خروجی تصویر است که خروجی بسیار مطلوبی برای کاربران این نوع دوربین است. طراحی مکانیکی این روبات بسیار ساده و ابتدایی و در عین حال مقاوم و مستحکم است. در آن از دو چرخ فعال در طرفین چپ و راست روبات و دو چرخ هرزه گرد در جلو و عقب ربات استفاده شده است که پایداری مناسبی را برای این ربات در سطوح صاف فراهم می‌کند. از بارزترین ویژگی‌های این روبات پشـتیبانی شـدن توسـط نرم‌افزار مشـهور و محبوب کاربران ربات‌ها به نام MRDS4 (Microsoft Robotics Developer Studio 4) و همچنین داشـتن سیستم عامل ROS (Robot Operating System) که امروزه اکثر روباتهای سکوی مشهور جهان مثل NAO با استفاده از این سیستم عامل برنامه ریزی می‌شوند.[۸]



شکل ۱-۷ روبات Eddie

## ۲-۱ معایب سکوهای کنونی

همانطور که در بخش قبل ذکر شـد، اغلب سـکوهای کنونی تک منظوره می‌باشند یا محدوده‌ی نسبتا کوچکی از کاربردها را حول یک نقطه‌ی خاص دارند. مانند MR Platform که از الگوریتمهایی خاصـی جهت شـناسایی توپ و مکان‌یابی آن و شناسایی روباتهای مجاور استفاده می‌کرد. همچنین سکوهای پیشرفته مانند نائو یا آسیمو یا روباتهای مانند آنهاکه همه منظوره می باشند قیمت‌های بسیار بالایی دارند و در حال حاضـر با توجه به وجود تحریم ها به آسـانی در اختیار همه ی محققان داخل کشور قرار نمی گیرند. به دلیل مشکلات روبات‌هایی که از حسگر اسـتفاده می‌کنند که در ادامه به آن اشـاره خواهد شد، در سال های اخیر مهاجرت از روباتهایی دارای حسگر به روبات های دارای دوربین زیاد شـده است. این طرز تفکر ایجاد شده است که به کمک پردازش تصویر، همه ی مشکلات روباتهای دارای حسگر حل شده است.[9] این نگرش بیشتر به پاک کردن صورت مسئله دلالت دارد تا حل مسئله. این دلایل باعث شده است تا نیاز به یک سکو همه منظوره ی ارزان و قابل اعتماد بیشتر احساس شود.

## ۱-۳ نیازسنجی برای داشتن یک Platform دارای حسگر چند منظوره بومی

امروزه عمده ترین دلیل عدم کارایی بالا در اجرای الگوریتم روبات‌های هوشمند غیر انسان‌نما که فقط از حسگر استفاده می‌کنند را می توان در موارد زیر خلاصه کرد:[۹, ۱۰]

### ۱-۳-۱ حسگرها

بیشتر روبات‌های هوشمند برای اجرای الگوریتم‌های خود نیاز به داشتن تعداد حسگرهای نسبتا زیادی دارند. در طراحی هر روبات محدودیت زیادی از جهت تعداد حسگرها وجود دارد. این محدودیت‌ها را می توان در موارد زیر خلاصه کرد:

- محدودیت مالی پروژه:

بودجه ی درنظر گرفته شده برای اجرای یک پروژه‌ی روباتیک اغلب محدود و مشخص می‌باشد. در بسیاری از موارد رقم بودجه محدود کننده‌ی اصلی تعداد حسگرهای یک روبات هوشمند می باشد.

- محدودیت پردازشی اطلاعات:

تعداد حسگرهای زیاد موجب افزایش بار پردازشی پردازنده‌ی روبات می شود. افزایش بار پردازشی به نوبه‌ی خود موجب افزایش توان مصرفی و کاهش سرعت پردازش خواهد شد. ضمن آنکه همیشه، همه‌ی اطلاعات بدست آمده از همه‌ی حسگرها مفید نیستند. بنابراین پردازنده‌ی اصلی باید پس از پردازش اطلاعات تک تک حسگرها، با توجه به شرایط پیش آمده انتخاب کند که اطلاعات کدام حسگر برای ادامه ی اجرای الگوریتم مفید می باشد. این مسئله به نوبه‌ی خود باعث تاخیر در اجرای الگوریتم اصلی روبات می‌شود.[۹]

- محدودیت مکانی:

هر حسگر مساحت مشخصی از فضای روبات را اشغال می‌کند. بنابراین یک عمل محدود کننده‌ی دیگر اما شاید با اهمیت نسبتا پایین‌تر در مقایسه با سایر موارد محدودیت فضای روبات می باشد.

کاهش تعداد حسگرهای یک روبات موجب می‌شود سخت‌افزار روبات نتواند تصویر دقیقی از محیط در اختیار پردازنده قرار دهد. بنابراین با کاهش تعداد حسگرهای یک روبات گرچه در مصرف توان صرفه جویی می‌شود و سرعت پردازش اطلاعات نسبتا افزایش می‌یابد، اما کارایی الگوریتم به نسبت کاهش خواهد یافت.[۹]

### ۱-۳-۲ موقعیت مکانی حسگرها

در بسیاری از روبات‌های هوشمند، حتی با وجود تعداد نسبتا زیاد حسگرها، باز هم کارایی الگوریتم چندان خوب نمی‌باشد. دلیل عمده‌ی این امر، عدم قرار گیری حسگرها در مکان‌های مناسب روبات می‌باشد. به این دلیل حسگرها نمی توانند محیط را به خوبی پوشش داده و اطلاعات خود را با دقت هر چه بیشتر در اختیار قرار دهند. به عبارت دیگر روبات بازهم در تهیه‌ی یک تصویر مناسب از محیط ناتوان است. در نتیجه روبات قادر به اجرای دقیق الگوریتم خود نیست.[۹]

### ۱-۳-۳ محدودیت‌های سخت افزاری پردازش

تمامی پردازنده‌ها محدودیت فضای آدرس دهی ورودی/خروجی دارند. برای مثال میکروکنتلر ATMega 32 که یکی از محبوب‌ترین میکروکنتلرها دربازار تجاری ایران می باشد، دارای ۴ پورت با کاربرد ورودی/ خروجی است که هر کدام از آنها ۸ پایه دارند.[۱۱] بنابراین در مجموع ۳۲ ورودی/خروجی تک پایه دارد. در نتیجه این پردازنده در حالت کلی حداکثر از ۳۲ حسگر ورودی پشتیبانی می‌کند. همچنین پردازنده‌ها علاوه بر محدودیت فضای آدرس‌دهی ورودی/ خروجی دارای محدودیت پروتکل نیز می‌باشند. برای مثال ATMega 32 از یک ارتباط سریال، سه تایمر و یک ارتباط دو سیمه (I2C) پشتیبانی می‌کند. البته قطعا می‌توان با

روش‌های مختلف همانند سوئیچینگ سخت افزاری یا نرم افزاری فضای آدرس‌دهی ورودی/خروجی یا حتی پروتکل‌ها را افزایش داد. اما این موضوع باعث افزایش مدارات واسط، افزایش توان مصرفی و کاهش سرعت پردازش خواهد شد. [۹]

### ۱-۳-۴ محدودیت توان

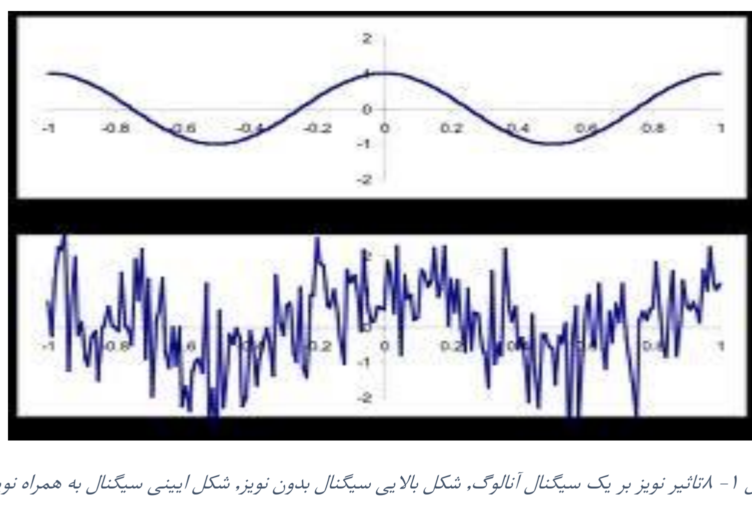
هر حسگر به صورت مجزا توان مصرف می‌کند. بنابراین با افزایش تعداد حسگرها مصرف توان افزایش می‌یابد. برای مثال اگر هر حسگر ۳۰ میلی آمپرجریان مصرف نماید و ۵ ولت اختلاف پتانسیل برای روشن شدن نیاز داشته باشد. ۱۵۰ میلی وات توان مصرف می‌نماید. حال اگر تعداد حسگرهای یک روبات ۱۰ عدد باشد (که غالبا رقم معقولی است) توان مصرفی روبات با صرفه نظر کردن از مصرف توان سایر قسمت ها حداقل برابر ۱۵۰۰ میلی وات می‌باشد که این میزان جریان در بعد الکترونیک توان زیادی است.[۹]

### ۱-۳-۵ افزایش تاخیر

پردازنده برای اجرای الگوریتم خود ابتدا باید اطلاعات ورودی از حسگرها را پردازش نماید. به عبارتی دیگر پردازنده ابتدا باید توابع مربوط به پردازش اطلاعات هر حسگر را فراخوانی نماید. بدیهی است اجرای هر تابع دارای تاخیر زمانی خاص خود است. بنابراین افزایش تعداد حسگرها موجب افزایش تاخیر زمانی خواهد شد.[۹]

### ۱-۳-۶ نویز

نویز یک سیگنال ناخواسته‌ی محیطی با دامنه‌ی کم می‌باشد که از سیگنال‌های الکتریکی و الکترومغناطیسی اطراف ناشی می‌شود. نویز باعث ایجاد اختلال در شکل سیگنال می‌شود. درنتیجه تشخیص سطوح سیگنال و پردازش آن با مشکل روبرو خواهد شد. شکل ۱-۸ تاثیر نویز را در به هم ریختن شکل یک سیگنال و سطوح آن نشان می‌دهد. نویز قابل تشخیص است اما قابل اندازه‌گیری نیست. با افزایش تعداد حسگرهای روبات، خطوط آنالوگ روبات نیز افزایش یافته و در نتیجه نویزپذیری سیستم افزایش می‌یابد. برای مثال هر چه تعداد حسگرها بیشتر باشد، تعداد گذرگاه‌های اطلاعاتی در روی مدار بیشتر است و این گذرگاهای اطلاعاتی علاوه بر این که خود یک منبع نویز می‌باشند، تحت تاثیر نویزهای جانبی محیط نیز قرار می‌گیرند. این موضوع باعث می‌شود اطلاعات صحیح در بسیاری از موارد به پردازشگر نرسد.[۹]



*شکل ۱ - ۸تاثیر نویز بر یک سیگنال آنالوگ. شکل بالایی سیگنال بدون نویز. شکل ایینی سیگنال به همراه نویز*

**نتیجه:** با توجه به ۶ مورد ذکر شده در بالا، یک رقابت بسیار نزدیک میان تعداد حسگرها، صحت و اعتبار اطلاعات، مصرف توان و سرعت پردازش وجود دارد. همچنین با کاهش تعداد حسگرها، روبات تصویر دقیقی از محیط خود در اختیار پردازنده قرار نمی‌دهد و در نتیجه کارایی الگوریتم پایین می‌آید. برای افزایش کارایی الگوریتم، نیاز به روباتی است که بتواند تصویری هرچه دقیق‌تر از محیط اطراف خود ارائه نماید. [۱۰]

## ۱-۴ ارائه‌ی معیارهای طراحی

با مطالعه و بررسی مقالات و همچنین گزارشات فنی و مستندات (فیلم، عکس) روبات‌های مین‌یاب، جستجوگر، نقشه‌بردار و میکروموس این نتیجه حاصل شد که با وجود طراحی و پیاده‌سازی الگوریتم‌های قوی، پردازنده‌ی این روباتها نمی تواند الگوریتم خود را به خوبی اجرا نماید. این امر ناشی از این است که حسگرهای روبات نمی‌توانند اطلاعات درستی از محیط در اختیار پردازنده قرار دهند.[۹] دلیل انتخاب روباتهای مین‌یاب، نقشه‌بردار، جستجوگر و میکروموس، پوشش دادن بخش بزرگی از مجموعه ماموریت‌های کلیدی روباتیک بود. روبات مین‌یاب، نقشه‌بردار و جستجوگر ماموریت‌های جستجو، نقشه‌برداری و جاروب‌کردن را انجام می دهد و روبات میکروموس نیز این ماموریت ها را با بررسی بسیار دقیق محیط پیاده سازی می نماید.[۱۲] بنابراین نیاز به یک روبات احساس می شود که بتواند بهترین، دقیق‌ترین، پیوسته‌ترین و کامل ترین تصویر را از محیط اطراف خود تهیه نماید. تصور می‌شود که این روبات باید بتواند شرایط زیر را ارضا نماید:

### ۱-۴-۱ صرفه جویی در هزینه ساخت

یکی از مهمترین دلایل اجرایی شدن یک پروژه به صرفه بودن آن از نظر اقتصادی است. بنابراین کم هزینه بودن روبات قطعا یکی از مهمترین ویژگی‌های آن است.

### ۱-۴-۲ انعطاف پذیری

امروزه طراحی سیستم‌های هوشمند به خصوص روبات‌های تحقیقاتی، به سمت چند منظوره بودن آن‌ها پیش می‌رود. به گونه‌ای که یک سیستم سخت افزاری واحد، باید به گونه‌ای از نظر نرم‌افزاری انعطاف‌پذیر باشد که بتواند برای مثال جایگزین مناسب برای اجرای الگوریتم‌های پایه مانند:جستجو، نقشه‌برداری، جاروب‌کردن و مانند این‌ها شود.

### ۱-۴-۳ سکوی قابل گسترش بودن

روبات باید این قابلیت را داشته باشد که بتواند دامنه‌ی نسبتا وسیعی از الگوریتم‌ها را بدون هیچ گونه تغییر بنیادین سخت افزاری پوشش دهد. منظور از تغییر بنیادین، تغییر کلی در رفتار الگوریتم است. روبات باید بتواند مشکل جمع‌آوری اطلاعات را حل کند، بدون آنکه به بخش‌های دیگر الگوریتم تغییری وارد کند. به عبارتی دیگر روبات باید یک سکوی قابل گسترش با قابلیت پوشش‌دهی گستره‌ی وسیعی از الگوریتم‌ها از جمله الگوریتم‌های جاروب کردن، جستجو، نقشه برداری و غیره باشد.

### ۱-۴-۴ حداقل تعداد حسگرها

تعداد حسگرهای روبات بایدکمترین مقدار ممکن باشد، به این معنی که به طور کلی روبات بتواند با استفاده از حداقل تعداد حسگر تصویر دقیق، کامل و پیوسته را از محیط بدهد و الگوریتم‌های خود را به خوبی و به طور دقیق با استفاده از این اطلاعات اجرا کند. زیرا همانطور که دربخش ۱-۳-۱ بحث شد، افزایش تعداد حسگرها باعث افزایش توان، کاهش کارایی الگوریتم و افزایش تاخیر سیستم می شود.

### ۱-۴-۵ ابعاد فیزیکی مناسب

ابعاد روبات باید معقول و مناسـب با عملیات تحت پوشـش باشـد. یعنی مطابق با نیاز ما و هم‌اندازه با سـایر روبات‌های همکار ساخته شـود. به این معنی که بتواند با شـرایط مختلف عملیاتی سازگار باشد. برای مثال روبات باید بتواند علاوه بر اینکه الگوریتم‌های یک روبات مین یاب هوشمند را در بعد کاملا عملیاتی اجرای می‌کند (میدان مین)، جایگزین یک روبات میکروموس برای حل یک ماز عملیاتی نیز باشد. منظور عملیاتی شـــرایطی نزدیک به محیط طبیعی با در نظر گرفتن تمام اختلالات محیطی اســـت. برای مثال یک ماز عملیاتی ممکن اســـت دارای مسیرهایی با عرض یکسان نباشد.[۱۰, ۱۳]

### ۱-۴-۶ بهره گیری از بیشترین سرعت پردازش

یکی از عوامل شـکسـت روبات‌ها در اجرا کردن الگوریتم خود، سرعت پائین پردازش اطلاعات است. این نیاز حس می‌شود که پردازش اطلاعات با بیشترین پالس ساعت انجام شود. منظور از "بیشترین پالس ساعت ممکن" حداکثر سرعت پردازنده با توجه به شرایط محدود کننده از جمله توان، طول عمر مفید روبات و عملیاتی بودن این سـرعت می‌باشد. منظور از عملیاتی بودن آن است که روبات بتواند به صورت طولانی مدت بدون هیچ گونه مشکلی در این سرعت پردازشی کار کند.

### ۱-۴-۷ طراحی مدار ماژولار

مدار روبات سـاده و قابل انعطاف باشـد. طراحی مدار روبات باید به صـورت ماژولار انجام شود. زیرا هدف از طراحی چنین روباتی چند منظوره بودن و ماهیت Platfrom آن اسـت. بنابراین مدار روبات باید تا حد زیادی قابل انعطاف باشد. هر بخش باید به صورت مجزا قابل تغییر، عیب‌یابی و گسترش باشد.

### ۱-۴-۸ سادگی و سرعت سخت افزار

در طراحی روبات به صـورت چند منظوره با ماهیت Platform، رقابتی جدی میان مصرف توان، ابعاد مدار، سادگی و سرعت مدار وجود دارد. روبات می‌تواند با حداقل واحدهای الکترونیکی سـاخته شود. در این حالت توان روبات پایین است، اما چند منظوره بودن آن نادیده گرفته می‌شود. بدیهی است که در این حالت به دلیل تمرکز بار پردازشی روبات روی حداقل تعداد پردازنده‌ها، سرعت پایین می‌آید. این شرایط با چند منظوره بودن روبات در تضاد است. در نتیجه سادگی و سرعت مدار در اولویت قرار می‌گیرد.

### ۱-۴-۹ بومی بودن

بعد از مشـخص شدن تک تک جنبه ها و ویژگی‌های روبات، یکی از مهمترین مسایل موجود در طراحی آن مطرح می‌شود. جایگاه این سـکو در میان صـدها سـکوی مشـابه در جهان چیست؟ پاسخ این سوال بومی بودن آن است. این سکو هرچه باشد یا هر امکاناتی داشته باشد، بدون بومی بودن فناوری آن امتیاز خاصی ندارد. بومی بودن در اینجا به معنای موجود بودن فناوری ساخت روبات در داخل کشور، سازگار بودن آن با شـرایط تحقیقاتی و عملیاتی داخل کشـور و برطرف کردن بخشـی از نیازهای موجود است. همانطور که در بخش ۱-۴ ذکر شد با بررسی مسـتندات روباتهای مین‌یاب، میکروموس و سـایر روباتهای مشـابه داخل کشـور، این نتیجه حاصـل شد که باوجود الگوریتم‌های نسبتا کامل و بهینه، سـخت افزار این روبات‌ها معایب فراوانی دارد. این معایب باعث می شود کارایی الگوریتم آنها کاهش یابد. همچنین مشخص شد که ایراد اصـلی این روبات‌ها عدم توانایی آنها در ایجاد تصـویری دقیق و پیوسـته از محیط اسـت. با توجه به بخش های ۱-۴-۱ تا ۱-۴-۸ یک سـکوی

ایده‌آل که برای رفع این مشکل طراحی می شود, باید بومی‌سازی شده باشد. زیرا در درجه ی اول، این یک مشکل داخلی و بومی است. هدف از طراحی چنین سکویی باید کاهش نیاز به نمونه های مشابه خارجی نیز باشد.

## ۱-۵ جمع بندی فصل اول

در این فصل در ابتدا به معرفی مفهوم Platform پرداخته شد. سپس چند نمونه از مشهورترین سکوها معرفی گردید. در ادامه به مشکل سکوهای غیرانسان نمای دارای سنسور در جمع آوری اطلاعات پرداخته شد. با بررسی جوانب این مشکل در داخل کشور، نیاز به یک سکو برای پوشـش دادن الگوریتم‌های نقشـه برداری، جاروب کردن و جسـتجو و همچنین الگوریتم‌های مشـابه آنها احساس شد. سپس به بررسی و طرح معیارهای طراحی این سکو پرداخته شد. در فصل آینده به ایده‌پردازی و روش های طراحی این سکو پرداخته می شود.

# فصل دوم

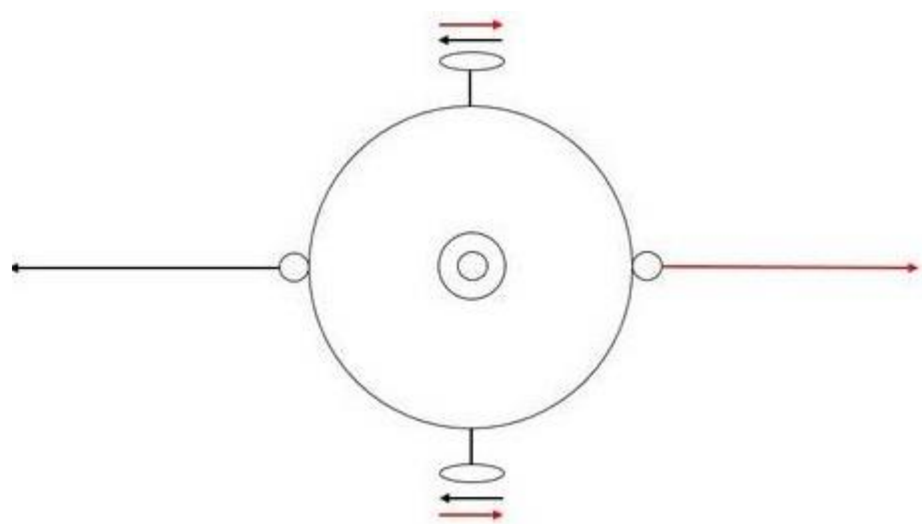
# انتخاب سخت‌افزار و نرم‌افزار روبات سکو

## ۲-۱ طرح کلی ایده‌ها

در این بخش، ایده‌ها به صورت کلی و بدون وارد شدن به جزئیات فروان مطرح می شوند. هدف از این بخش، ایجاد یک دور نمای کلی از طرح‌های بررسـی شده در حین پژوهش بر ساختار فرض مطرح شده در فصل اول می‌باشد. جزئیات هر ایده و همچنین مزایا و معایب آنها به طور کامل در بخش‌های بعدی همین فصـل بررسـی خواهند شـد. در پالایش ایده‌های مختلف و با توجه به فرض های مطرح شـده در فصـل اول،این نتیجه حاصـل شد برای ارضا کردن فرض‌های حداقل تعداد حسگر و همچنین ایجاد تصویر پیوسته و کامل از محیط، روبات سکو مورد نظر باید دارای یک یا چند حسـگر با قابلیت چرخش و داشـتن حداقل یک درجه ی آزادی باشد. از این رو تمامی ایده‌هایی که در ادامه مطرح می‌شوند، با درنظر گرفتن این فرضیه مطرح می‌شوند.

### ۲-۱-۱ ایده شماره ۱

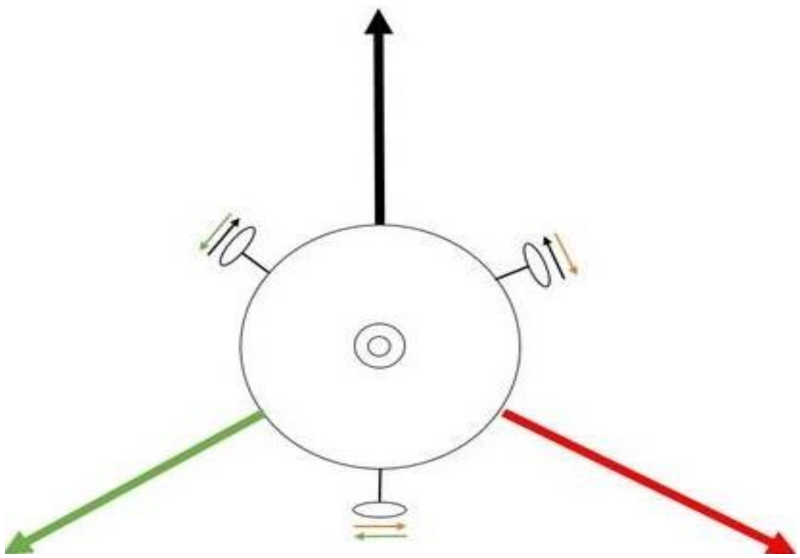
مطابق شکل شماره ۲-۱ اولین ایده‌ی طرح شده یک روبات دارای دوچرخ متصل به موتور گریبکس دار به همراه دو چرخ ثابت بود. همچنین از یک موتور برای گردش یک حسگر در بالای روبات استفاده شده بود.



*شکل ۲- ۱ طرح کلی ایده شماره ۱، هریک از دو دایره بالا و پایین روبات نمایانگر چرخهای حرکتی متصل به موتورها می‌باشند، همچنین دایره های سمت چپ و راست روبات معرف چرخهای هرزگرد آن است. دو دایره‌ی کوچک وسط روبات حسگر و موتور گرداننده‌ی حسگر را نشان می‌دهند.*

### ۲-۱-۲ ایده شماره ۲

در این ایده روبات دارای ۳ چرخ خورشیدی که به صورت مستقیم به سه موتور گریبکس‌دار متصل شده اند، درنظر گرفته شد. همانطور که در شکل ۲-۲ مشاهده می شود، در این ایده فرض شد که موتورهای روبات با یکدیگر زاویه ی ۱۲۰ درجه می‌سازند. همچنین همانطور که در ایده‌ی شماره ۱ مطرح شد، فرض می شود یک یا دو حسگر بر روی روبات قرار داده می شود. این حسگر دارای یک درجه‌ی آزادی بوده و توسط یک موتور به گردش در می‌آید.



*شکل ۲- ۲ طرح کلی ایده شماره ۲، هریک از دایره هایی که در سه طرف روبات با زاویه‌ی ۱۲۰ درجه نسبت به یکدیگر قرار گرفته‌اند، یکی از چرخهای خورشیدی روبات را نشان می‌دهند، همچنین دو دایره‌ای که در وسط روبات قرار گرفته‌اند، معرف حسگر و موتور گرداننده‌ی حسگر است.*

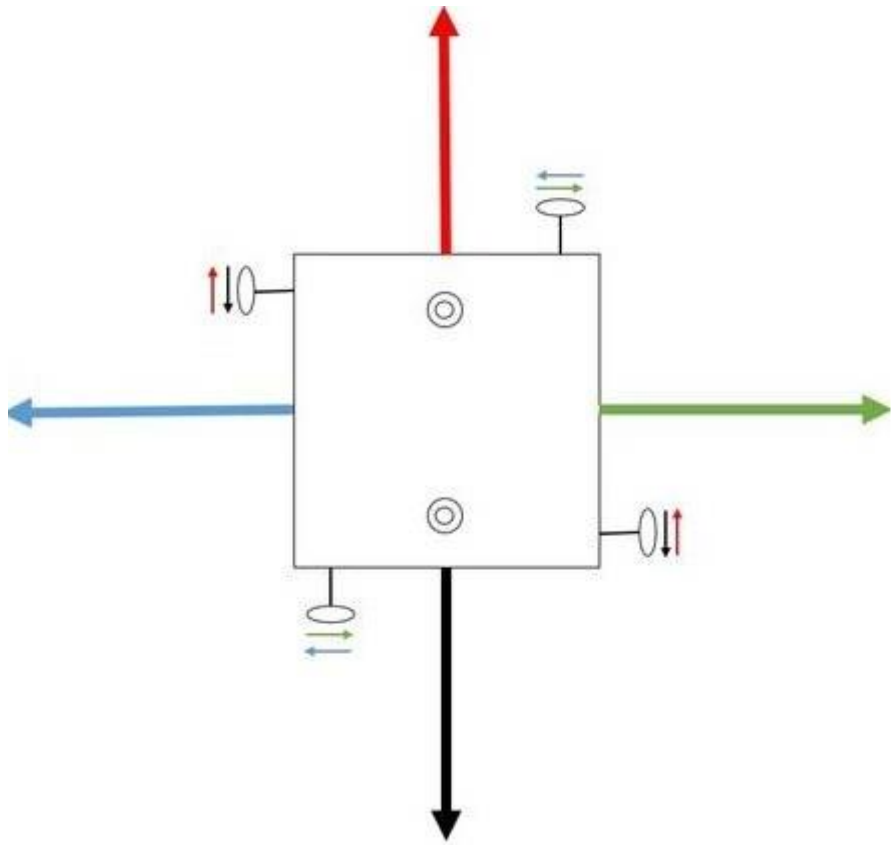
همانطور که در شکل شماره ۲-۳ مشاهده می‌شود، چرخ‌های خورشیدی این قابلیت را دارند که علاوه بر چرخش و حرکت حول محور مرکزی خود، در راستای عمود بر چرخ نیز حرکت داشته باشند.



*شکل ۲- ۳ چرخ خورشیدی*

### ۲-۱-۳ ایده شماره ۳

ساختار کلی ایده شماره ۳ مانند ایده‌ی شماره ۲ می باشد، با این تفاوت که به جای ۳ چرخ خورشیدی با زاویه ۱۲۰ درجه با یکدیگر، از ۴ چرخ خورشیدی با آرایش شکل شماره ۲-۴ استفاده شده است. همچنین مطابق سایر ایده‌ها از یک یا دو حسگر که بر روی موتور قرار دارند و دارای یک درجه‌ی آزادی می‌باشند استفاده خواهد شد.



*شکل ۲- ۴ طرح کلی ایده شماره ۳، چهار دایره‌ی کوچک در چهار سمت روبات معرف چرخهای حرکتی روبات هستند. این چرخها همگی خورشیدی می‌باشند. زاویه‌ای که چرخهای راستای افق با چرخهای راستای قائم می‌سازند ۹۰ درجه است. دایره هایی که در داخل روبات در سمت بالا و پایین قرار دارند، حسگرها و موتورهای گرداننده‌ی حسگر را نشان می‌دهند.*

## ۲-۲ جنس شاسی روبات

برای جنس ماده‌ی تشکیل دهنده ی شاسی، سه گزینه‌ی زیر مطرح شد:

- پلکسی گلاس: پلکسی گلاس نوعی پلاستیک می‌باشد که ظاهری بسیار شفاف و شبیه به شیشه دارد. جنس پلکسی گلاس از نوعی پلی‌کربنات شفاف می‌باشد که نوعی پلیمر است. از دیگر ویژگی‌های پلکسی، مقاومت زیاد آن نسبت به شیشه است.
- آلومینیوم: آلومینیوم بسیار سبک در عین حال دارای مقاومت نسبتا بالا است.
- تفلون(نوعی پلاستیک):این ماده نوعی پوشش نچسب و عایق الکتریسیته است که هم مقاومت حرارتی بالا و هم مقاومت شیمیایی خوبی در مقابل اسیدها و بازها دارد.

## ۲-۳موتورهای پیشنهادی

موتورهای پیشنهادی برای پیاده‌سازی ایده‌های مطرح شده، به دو بخش موتورهای حرکتی روبات و موتورهای گرداننده‌ی حسگرها تقسیم‌بندی می‌شوند. وظیفه‌ی موتورهای حرکتی روبات، حرکت دادن و جابجایی فیزیکی تمامی بدنه و شاسی روبات است. موتورهای گرداننده‌ی حسگر فقط وظیفه‌ی چرخش و گردش حسگرها را بر عهده دارند.

### ۲-۳-۱ موتورهای حرکتی روبات

- موتور DC: موتور DC، نوعی ماشین الکتریکی است که الکتریسیته را به حرکت مکانیکی تبدیل می‌کند. اکثر موتورهای الکتریکی توسط الکترومغناطیس کار می‌کنند، اما موتورهایی که بر اساس پدیده‌های دیگری نظیر نیروی الکترواستاتیک و اثر پیزوالکتریک کار می‌کنند، هم وجود دارند. اغلب موتورهای الکتریکی دوار هستند، اما موتور خطی هم وجود دارند. در یک موتور دوار بخش متحرک (که معمولاً درون موتور است) چرخانه یا روتور و بخش ثابت ایستانه یا استاتور خوانده می‌شود. موتور شامل آهنرباهای الکتریکی است که روی یک قاب سیم پیچی شده‌است. گر چه این قاب اغلب آرمیچر خوانده می‌شود، اما این واژه عموماً به غلط بکار برده می‌شود. در واقع آرمیچر آن بخش از موتور است که به آن ولتاژ ورودی اعمال می‌شود یا آن بخش از ژنراتور است که در آن ولتاژ خروجی ایجاد می‌شود. با توجه به طراحی ماشین، هر کدام از بخش‌های چرخانه یا ایستانه می‌توانند به عنوان آرمیچر باشند. برای ساختن موتورهایی بسیار ساده کیتهایی را در مدارس استفاده می‌کنند. [۱۴]
- موتور پله ای (Stepper): موتور پله‌ای یک موتور الکتریکی می‌باشد که حرکت آن کاملا دقیق و از پیش تعریف شده است. این موتور ورودی الکتریکی دیجیتال را به یک حرکت مکانیکی تبدیل می‌کند و با ارسال بیت‌های ۰،۱ به سیم پیچهای آن می‌توان آن را حرکت داد. در واقع یک موتور پله ای ترکیبی از یک موتور الکتریکی DC و یک سلونویید است. موتورهای پله‌ای ساده توسط بخشی از یک سیستم دنده ای در حالت‌های موقعیتی معین قرار می‌گیرند، اما موتورهای پله‌ای نسبتا کنترل شده، می توانند بسیار آرام بچرخند. موتورهای پله‌ای کنترل شده با کامپیوتر یکی از فرم‌های سیستم‌های تنظیم موقعیت هستند. این موتورها به صورت درجه‌ای دوران می کنند و با درجه‌های مختلف در بازار موجود هستند. واژه‌ی پله در اینجا, به معنی چرخش به اندازه درجه تعریف شده موتور است. مثلا موتور پله‌ای با درجه ۱,۸ باید ۲۰۰ پله انجام بدهد تا ۳۶۰ درجه یا یک دور کامل بچرخد (۱,۸×۲۰۰ =۳۶۰ ) و یک موتور پله‌ای با درجه ۱۵ فقط باید ۲۴ پله برای انجام یک دور کامل بچرخد (۲۴ ×۱۵ =۳۶۰ ) . این ویژگی مزایای بسیار زیادی برای مصرف کننده دارد. از جمله‌ی این فواید، امکان کنترل سرعت و موقعیت دقیق موتور پله‌ای می باشد.

این موتور عموما دارای چهار قطب است. سیم‌پیچ‌ها بر روی این چهار قطب قرار می گیرند. کاربر با ارسال بیتهای ۰و۱به این سیم‌پیچ‌ها در واقع میدان مغناطیسی ایجاد می‌کند. این میدان باعث حرکت روتور مغناطیسی موجود در داخل موتور پله‌ای می‌شود. این سیم پیچ‌ها را باید به توالی مشخصی ۰ و ۱ کرد. در غیر این صورت، موتور مطابق تصور کاربر نخواهد چرخید. یکی از مشخصه‌های این موتور زاویه حرکت آن می‌باشد. هر موتوری زاویه حرکتی مخصوص به خودش را دارد. برای مثال، اگر موتوری زاویه حرکتش ۷درجه باشد، در هر باری که سیم پیچهایش حاوی ولتاژ می شوند، ۷ درجه در سمت حرکت عقربه‌های ساعت یا خلاف جهت آن بسته به اینکه سیم‌پیچ‌ها با چه ترتیبی تحریک می شوند، خواهد چرخید. این ۷ درجه چرخش برای این موتور پله‌ای نمونه‌ی یک پله یا یک step محسوب می شود. پس یک موتور پله‌ای در یک دور کامل ممکن است ۱۰۰تا ۲۰۰ پله کمتر یا بیشتر بسته به نوع موتور داشته باشد.

این موتور به صورت یک بیتی یا دو بیتی حرکت می‌کند. در حالت یک بیتی در هر لحظه تنها یک سیم پیچ پالس یک را دریافت می‌کند. در حالت دو بیتی دو سیم پیچ در هر لحظه پالس یک را دریافت می کنند. اگر این دریافت پالس به صورت منظم و پشت سر هم انجام شود، موتور نیز به صورت صحیح به سمت جهت حرکت عقربه های ساعت یا خلاف جهت آن حرکت خواهد کرد.[۱۴]

### ۲-۳-۲ موتورهای گرداننده‌ی حسگر

برای گردش حسگر موتور DC پیشنهاد نمی شود. زیرا کنترل حرکت آن در حالت کلی دقیق نیست. دقت بالا در کنترل حرکت موتور DC، مستلزم پیاده‌سازی محاسبات مکان، سرعت و شتاب و همچنین مدارات سخت‌افزاری خاصی جهت شمارش می‌باشد. بنابراین در بررسی‌های اولیه این نوع موتور برای گرداندن حسگر حذف شد و موتورهای سرو و پله‌ای به عنوان موتورهای گرداننده‌ی حسگر پیشنهاد شدند.

- سرو موتور (Servo Motor): سرو موتور در واقع یک موتور DC به همراه یک مدار کنترل‌کننده است. برخلاف موتورهای DC و پله‌ای که با تغییر سطح ولتاژ تحریک می شوند, سرو موتور بوسیله ی یک پالس با Duty Cycle مشخص تحریک می شود. تحریک سرو موتور به صورت چرخیدن یک یا چند دور کامل نیست، بلکه به صورت حرکت از زاویه‌ی صفر به یک زاویه بین ۹۰- درجه تا ۹۰+ درجه می باشد. هرچه مدت زمان یک بودن سیگنال بیشتر باشد، سرو موتور به زاویه‌ی بزرگتری خواهد رفت. وقتی سرو موتور به مرزهای بازه‌ای خود برسد، با افزایش یا کاهش Duty cycle تغییری نخواهد داشت. برای آنکه سرو موتور در یک زاویه خاص بماند، پالس تحریکی باید ادامه داشته باشد. [۱۵]

## ۲-۴ ایده‌های پیشنهادی برای پیاده‌سازی بدنه‌ی کلی سخت افزار

در این بخش ایده‌های پیشنهادی برای پیاده‌سازی بخش‌های مختلف سخت‌افزار روبات مطرح می‌شوند، از جمله‌ی این بخش‌ها می‌توان به سیستم پردازشی، طراحی مدار و شیوه‌ی پردازش اطلاعات اشاره کرد.

### ۲-۴-۱ ایده‌های پیشنهادی سیستم پردازشی

ایده‌های پیشنهادی سیستم پردازشی در زمینه‌ی شیوه‌ی پیاده‌سازی سخت افزاری سیستم پردازشی روبات می باشد. این ایده‌ها شامل دو زیرگروه هستند: پردازش توزیع شده، پردازش متمرکز

#### پردازش توزیع شده

در این نوع پردازش اطلاعات، هر بخش توسط یک ریزپردازنده پردازش می‌شود. همه‌ی بخش‌ها با یکدیگر در ارتباط هستند. یکی از جنبه‌های مثبت این گونه طراحی این است که می‌توان تمامی قسمت‌ها را به طور جداگانه برنامه‌ریزی کرد و سرعت اجرای الگوریتم نیز بالا می‌رود. همچنین بار پردازشی کل بر روی چندین پردازنده تقسیم می‌شود. امکان موازی‌سازی پردازش‌ها بوجود می‌آید و از تاخیر به خاطر آماده نبودن اطلاعات یک یا چند بخش، جلوگیری می شود. در برابر این مزایا, این شیوه پردازشی مشکلاتی نیز به همراه دارد. از جمله ی این مشکلات می‌توان به بالا رفتن هزینه‌ی تولید سخت افزار مدار, افزایش مصرف توان و افزایش مساحت فیزیکی مدار اشاره کرد.[۹]

#### پردازش متمرکز

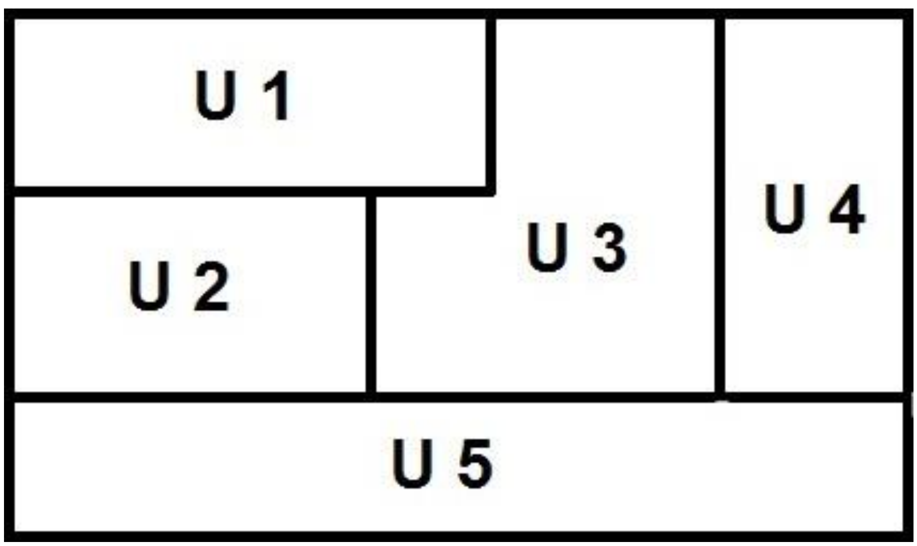
در این نوع طراحی پردازش اطلاعات تمامی بخش‌ها فقط توسط یک ریز پردازنده انجام می‌گیرد. نسبت به روش قبل این روش با سرعت بسیار کمتری الگوریتم را اجرا می‌کند. بار پردازشی در این روش، به طور کامل بر روی یک پردازنده قرار می گیرد. از ویژگی‌های این نوع پردازش می‌توان به سادگی مدار، کاهش مساحت فیزیکی مدار، کاهش مصرف توان و کاهش هزینه اشاره کرد.[۹]

### ۲-۴-۲ ایده‌های پیشنهادی طراحی مدار

ایده‌های طراحی مدار، شامل نحوه‌ی پیاده‌سازی بخش‌های مختلف مدار و ارتباط آنها با یکدیگر می‌باشد، این ایده‌ها شامل موارد زیر هستند:

#### طراحی ماژولار

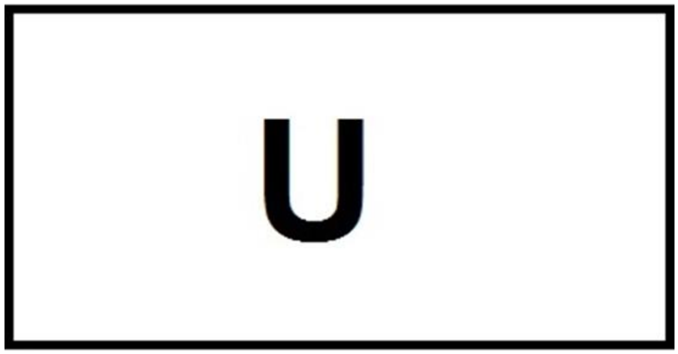
طراحی مدار ماژولار به معنی بخش بخش بودن قسمت‌های مختلف مدار است. هر بخش توسط یک یا چند ریزپردازنده کنترل می‌شـود. هر بخش به صـورت کاملا مستقل از سایر بخش‌ها بوده و به طور مجزا از سایر بخش‌ها تغذیه می‌شود. با وجود جدا بودن هر بخش از بخش دیگر تمـامی بخش‌هـا با یکدیگر در ارتباط و تامل هســـتند. از فواید اینگونه طراحی می‌توان به عیب‌یابی راحت مدار و تعویض و ارتقاء بخش‌های مدار اشـاره کرد. در مقابل ماژولار طراحی کردن موجب افزایش توان می‌شـود زیرا هر بخش سیستم تغذیه خاص خود را خواهد داشــت. همچنین مســاحت فیزیکی مدار و قیمت آن نیز افزایش می یابد. شــکل ۲- ۵ مثالی از ایده ی طراحی ماژولار را نشـان می‌دهد. همانطور که در این شـکل ملاحظه می‌شـود، سـیستم فرضی از ۵ ماژول تشکیل شده‌اند، این ماژول‌ها کاملا مستقل از یکدیگر می‌باشند. [۹]




*شکل ۲- ۵ طراحی ماژولار، در این طراحی، مرز شفاف مشخص و قابل رویت میان تمامی بخش های مستقل مدار وجود دارد.*

#### طراحی غیرماژولار

در طراحی مدار به صـورت غیرماژولار، قسمت‌های مختلف مدار از یکدیگر جدا نبوده و مدار به صورت یک بخش واحد است. همانطور که در شـکل شـماره ۶-۲ نیز ملاحظه می شـود، در این طراحی نمی‌توان بخش‌های مدار را به صورت کاملا قطعی و مستقل مشخص کرد. به عبارت دیگر هیچ‌گونه مرزبندی دقیقی میان بخش‌های مدار وجود ندارد.[۹, ۱۶]




*شکل ۲- ۶ طراحی غیر ماژولار، در این طراحی، هیچ گونه مرز مستقل میان بخش های مختلف مدار وجود ندارد و بخشهای از یکدیگر قابل تفکیک نیستند.*

### ۲-۴-۳ ایده‌های کلی برای پردازش اطلاعات

ایده‌های کلی پردازش اطلاعات شامل نحوه‌ی پیاده‌سازی نرم‌افزاری سیستم پردازش اطلاعات می‌باشند. این ایده‌ها به دو دسته‌ی پردازش الگوریتم روی مدار و پردازش الگوریتم توسط یک عامل خارجی تقسیم‌بندی می‌شوند.

- **پردازش الگوریتم توسط مدار**

در این روش، پردازش تمامی الگوریتم‌های روبات، توسط خود روبات انجام می‌شود. روبات برای پردازش اطلاعات خود، هیچ‌گونه ارتباطی با محیط بیرون خود ندارد. تمامی اطلاعات خام توسط پردازنده‌های روبات مورد پردازش قرار می‌گیرند. چون اطلاعات توسط ریزپردازنده‌های روبات مورد پردازش قرار می گیرند، غالبا سطح زبان برنامه نویسی مورد استفاده پایین یا حداکثر متوسط می‌باشد. به عبارتی دیگر پیاده‌سازی زبان‌های سطح بالا و شی گرا، نیازمند زیرساخت‌های قوی سخت‌افزاری بر روی روبات می‌باشد. این امکان به طور عام برای همه‌ی سیستم‌های سخت‌افزاری قابل استفاده نیست. [۹, ۱۶]

- **پردازش الگوریتم توسط عامل خارجی**

در این روش، اطلاعات خام توسط روبات جمع‌آوری شده و به یک عامل خارجی ارسال می‌شود. عامل خارجی اطلاعات را دریافت می‌کند و آنها را مورد پردازش قرار می‌دهد. سپس خروجی پردازش به صورت فرمان در اختیار روبات قرار می‌گیرد. منظور از پسوند "خارجی" در اینجا این است که مرز شفاف و دقیقی میان عامل و روبات می‌باشد. به طوری که روبات از عامل خارجی قابل تشخیص است و هردو به صورت کاملا مستقل از هم عمل می‌کنند. [۹]

## ۲-۵ ایده‌های پیشنهادی در مورد نوع حسگر روبات

مطابق با معیارهای ذکر شده در فصل اول، در بخش انتخاب نوع حسگرهای روبات با تنوع وسیعی روبرو هستیم. حسگرهایی با کاربرد اصلی تشخیص مانع یا ارزیابی فاصله تا یک مانع در بازه‌ی وسیعی از تنوع در بازار جهانی یافت می‌شوند. حسگرهایی مانند فراصوت، مادون‌قرمز، سونار، تماسی، لیزر و نظیر آنها، نمونه ای از حسگرهای موجود می‌باشند.[۱۵] با این وجود در نظر داریم دامنه‌ی ایده پردازی در مورد حسگرهای روبات را محدود کنیم. دلیل این موضوع نه به خاطر وجود ضعف یا ناتوانی خاصی در یک خانواده از حسگرها بلکه به دلیل یکی از معیارهای ۹ گانه‌ای بود که برای طراحی روبات در فصل ۱ عنوان شد. در واقع هریک از معیارهایی که در فصل اول برای طراحی این روبات سکو مشخص شدند، به منزله‌ی خط قرمزی برای ایده‌پردازی بودند و مرزی روشن را برای طراحی روبات مشخص کردند. با در نظر گرفتن تمامی این

معیارهای ۹ گانه، مرزهای اصلی این پروژه مشخص شدند. بدیهی است هرگونه ایده پردازی باید در محدوده‌ی مشترک داخل این مرزها صورت گیرد.

مطابق با معیار ۱-۴-۹ که بدون شک اصلی‌ترین معیار طراحی در این پروژه می‌باشد، Platform ساخته شده باید "بومی" باشد. همانطور که در بخش ۱-۴-۹ نیز گفته شد بومی بودن به معنای یافت شدن وسایل و فناوری ساخت آنها در بازار داخلی و به سهولت می‌باشد. همچنین به عنوان یک روبات بومی که می‌خواهد جایگزین بخش نسبتا وسیعی از روبات‌های بومی دیگر باشد، باید هزینه‌ی ساخت آن تا حد امکان تعدیل شود. به طوریکه هزینه‌ی نهایی پروژه حداقل و در بدبینانه‌ترین حالت، با میانگین قیمت روبات‌های مشابه خارجی برابر باشد. با در نظر گرفتن این دو موضوع (بومی بودن و کم هزینه بودن) در بخش ایده پردازی، دامنه ی حسگرهای مورد بررسی را به دو خانواده از حسگرهای محبوب، پرفروش و پرطرفدار در بازار داخلی کاهش می‌دهیم. این دو خانواده همچنین دارای ارزان‌ترین حسگرهای موجود در بازار و سهل الوصول‌ترین آنها می‌باشند.

### حسگر فراصوت

حسگرهای فراصوت بر مبنای ارسال و دریافت امواج فراصوت کار می کنند. این حسگرها به طور خاص برای تشخیص مانع و تشخیص فاصله از مانع استفاده می‌شوند. هر حسگر از یک فرستنده و یک گیرنده‌ی امواج فراصوت تشکیل شده است. مکانیسم کار این حسگرها بر این مبنا می‌باشد که فرستنده یک موج فراصوت با طول موج مشخص می‌فرستد. این موج به راه خود ادامه می‌دهد تا زمانی که به یک مانع برخورد می کند. زمانی که موج با مانع برخورد کرد، بخشی از آن توسط مانع جذب می‌شود و مابقی آن از مانع بازتاب می‌شود. زمانی که موج بازتاب شونده توسط گیرنده دریافت می‌شود، حسگر بوسیله‌ی اختلاف زمانی بین رفت و برگشت این موج و همچنین سرعت موج فراصوت، فاصله‌ی خود را از مانع محاسبه می نماید.[۱۵]

### حسگر مادون قرمز

حسگرهای مادون قرمز مبنای کاری نوری-گرمایی دارند. همه‌ی اجسام در دمای بالاتر از صفر درجه‌ی کلوین از خود امواج نورانی تابش می‌کنند. شدت این امواج در هر نقطه به رنگ و دمای تابش کننده و فاصله تا آن بستگی دارد. همانند حسگرهای فراصوت، این حسگرها نیز از یک فرستنده و یک گیرنده تشکیل شده‌اند. فرستنده پرتو مادون قرمزی را با طول موج مشخص می‌فرستد. این پرتو به مانع برخورد کرده، بخشی از آن توسط مانع جذب می‌شود و بخش دیگر آن از مانع بازتاب می‌شود. پرتو بازتاب شده به گیرنده می‌رسد. گیرنده با دریافت آن، اختلاف پتانسیلی متانسب با فاصله‌ی مانع از خود تولید می‌کند. با مقایسه این اختلاف پتانسیل با یک اختلاف پتانسیل مرجع و همچنین فرمول‌بندی آن، می توان فاصله‌ی جسم از حسگر را اندازه گرفت.[۱۵]

## ۲-۶ پالایش و ارزیابی ایده‌ها

در این بخش ایده‌های طرح شده از ابتدای این فصل مورد بحث و بررسی قرار می‌گیرند. معایب و مزایای آنها مشخص می‌شوند و ایده‌ی برتر برای ساخت Platform در هریک از بخش‌ها مشخص می‌شود.

### ۲-۶-۱ بررسی ایده‌های مکانیکی

- بررسی ایده شماره ۱:

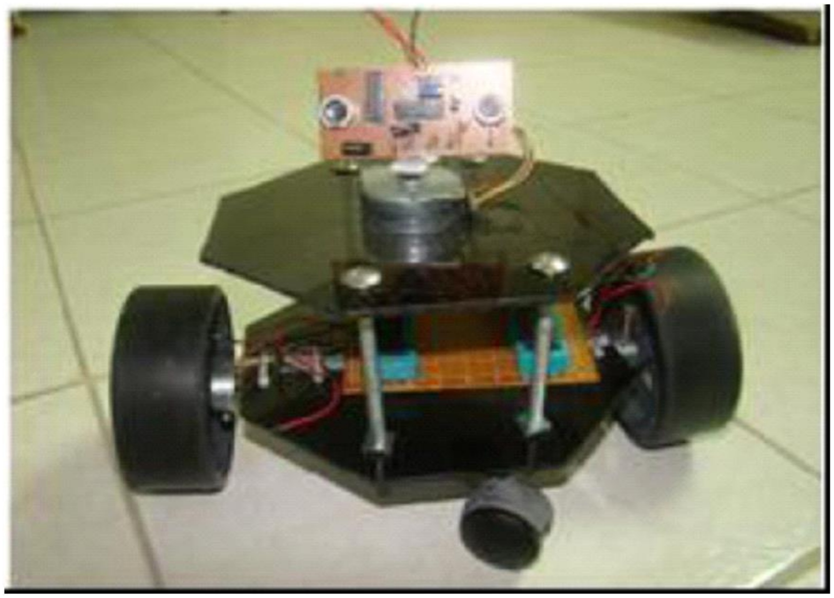
تصاویر شماره ۲-۷ و ۲-۸ نمایی از مکانیک ساخته شده‌ی ایده‌ی شماره ۱ را نشان می‌دهد. پس از ساخت ایده مکانیک شماره ۱, معایب و مزایای آن به صورت زیر مشخص شدند:

مزایا:

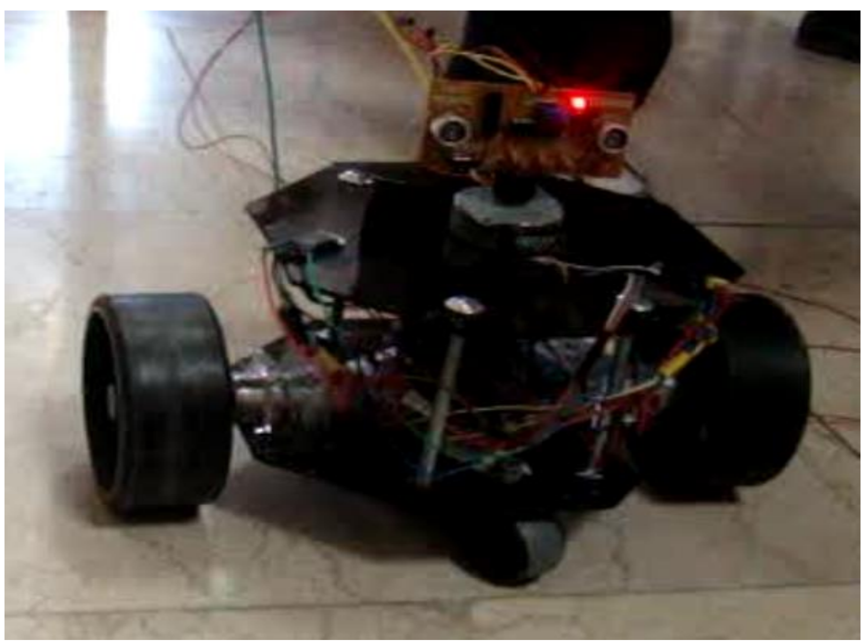
- سادگی فرآیند ساخت نسبت به سایر ایده‌ها
- نیازمند نبودن به دقت بالا در هنگام ساخت
- هزینه‌ی پایین فرآیند ساخت نسبت به سایر ایده‌ها
- روبات در دستگاه مختصات دکارتی مدل می‌شود. محاسبات در این دستگاه برای پردازنده‌ها نسبت به سایر دستگاه‌ها سریعتر است و بار پردازشی کمتری می‌برد.

معایب:

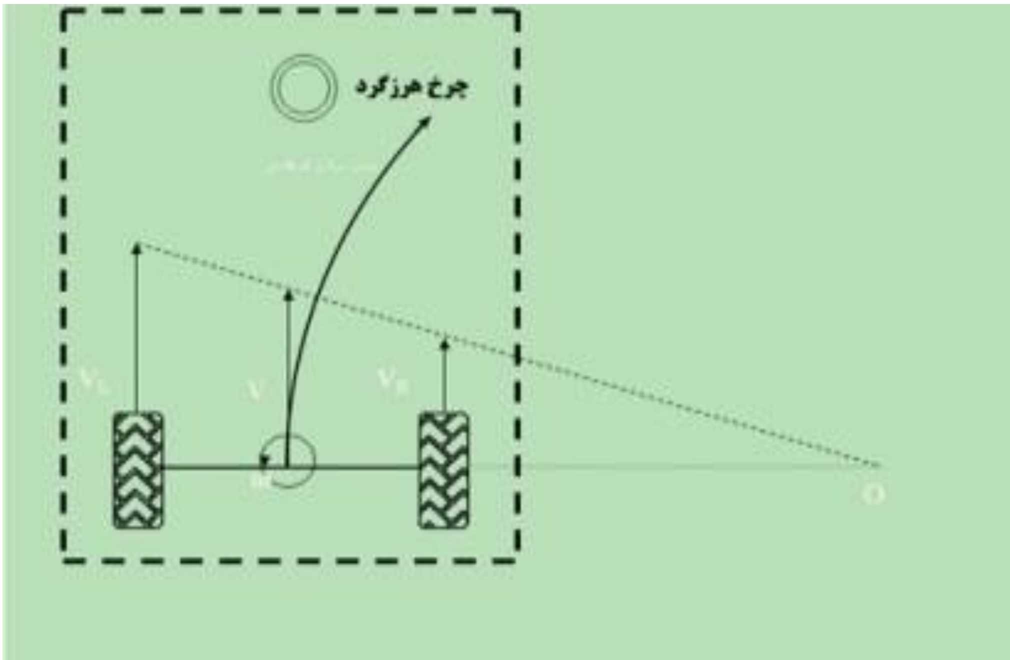
- مختصات دقیق مکان روبات بعد از دور زدن چرخ ها و تغییر جهت، از طریق محاسبات ساده ریاضی قابل محاسبه نمی‌باشد و نیازمند محاسبات پیچیده‌تر می‌باشد. این موضوع باعث می‌شود دقت روبات در مکان‌یابی کاهش یابد. همچنین بار محاسباتی زیادی بعد از هربار دور زدن بر روبات تحمیل می‌شود.
- دقت روبات در مکان‌یابی خود بعد از هربار دور زدن به صورت تصاعدی کاهش می‌یابد. دلیل این امر آن است که مطابق شکل ۲-۱۰ روبات با آرایش تفاضلی دور می‌زند. در این آرایش به هنگام دورزدن، یک چرخ با سرعت بیشتر و چرخ دیگر با سرعت کمتر می‌چرخد و روبات حول یک نقطه خاص دور می‌زند که به آن مرکز دوران روبات گفته می‌شود. همانطور که در پاراگراف قبل بیان شد، محاسبه دقیق مکان روبات در این حالت نیازمند محاسبات بسیار پیچیده ریاضی است. این امر باعث می شود که روبات در محاسبه مختصات خود بعد از دوران، حتی در بهترین حالت, دچار اندکی خطا شود. این خطا در دوران بعدی قدری افزایش می‌یابد. زیرا مختصات دوران قبلی دقیق نبوده است. این روند به صورت تصاعدی ادامه می‌یابد. همچنین چنانچه روبات در حین دور زدن دچار لغزش شود، مرکز دوران آن جابجا شده و محاسبات حول نقطه‌ی اشتباهی انجام می‌شود.[۱۷]



*شکل ۲- ۷ ساخت مکانیک ایده شماره ۱ این روبات به Mono Sensor Platform Robot یا به طور اختصار MSPR معروف شده است.*



*شکل ۲- ۸ نمایی دیگر از نمونه ی ساخته شده ی ایده شماره ۱، این روبات در دستگاه مختصات دکارتی مدل می‌شود و نسبت به سایر ایده ها هزینه‌ی ساخت کمتری دارد.*




*شکل ۲- ۹ نحوه دور زدن روبات در ایده شماره۱ ، در این ایده، هنگام دوردورزندن روبات، یک چرخ با سرعت بیشتر و چرخ دیگر با سرعت کمتر می‌چرخد، بنابراین روبات حول یک مرکز دوران O شروع به دورزدن می‌کند. محاسبه مکان روبات بعد از دورزدن در این روش مشکل است، خطاپذیری آن بیشتر است و نیازمند محاسبات دقیق‌تر است.*

- بررسی ایده‌ی شماره ۲:

تصاویر ۲-۱۰، ۲-۱۱ و ۲-۱۲ نمونه‌ی ساخته شده‌ی ایده‌ی شماره ۲ را نشان می‌دهند. پس از بررسی و انجام آزمایش بر روی این ساختار مکانیکی، معایب و مزایای آن به شرح زیر مشخص شدند:

مزایا:

- حرکت روبات در دستگاه مختصات استوانه‌ای مدل می‌شود. مطابق شکل ۲-۱۳ در این دستگاه برای محاسبه ی مختصات روبات در دو بعد در هر لحظه تنها به بردار جابجایی روبات از مبدا مختصات و زاویه‌ای که این بردار با مبداء مختصات می‌سازد نیاز است. در نتیجه دورزدن و تغییر جهت روبات در این دستگاه مختصات به سادگی، بدون هیچ جابجایی و فقط با تغییر زاویه $\theta$ انجام‌پذیر است. همانطور که گفته شد در هنگام دور زدن روبات، مولفه‌ی $r$ هیچ تغییری نمی‌کند. بنابراین روبات به هنگام دورزدن هیچ‌گونه جابجایی نخواهد داشت. بنابراین بر خلاف ایده شماره ۱، این ایده مشکل تعیین مختصات روبات بعد از جابجایی را به صورت کامل حل می‌کند.

معایب:

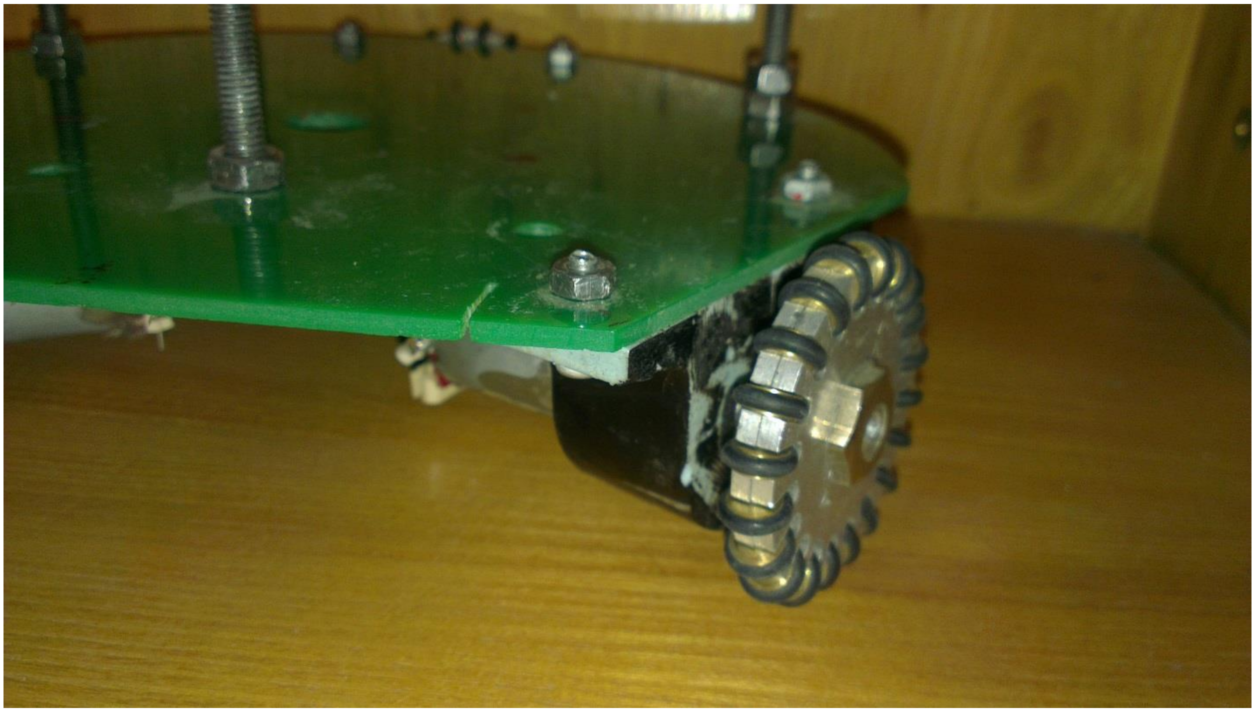
- این ایده به دقت نسبتا بالایی در هنگام ساخت نیازمند است، زیرا زاویه‌ی میان موتورهای آن باید ۱۲۰ درجه باشد. هرگونه انحراف از این زاویه موجب خطا در محاسبه‌ی مختصات حرکت روبات بعد از جابجایی خواهد شد.
- سرعت موتورها باید با هم برابر باشند. زیرا هر حرکت روبات به استثناء دور زدن آن مطابق شکل ۲-۱۴ به صورت دو بردار هم اندازه که با یکدیگر زاویه ۱۲۰ درجه می سازند مدل می‌شود. حال اگر

سـرعت یکی از چرخ‌ها به هنگام حرکت از دیگری بیشـتر شود، مطابق شکل ۲-۱۵ بردار جابجایی دارای انحراف خواهد بود و بنابراین محاسبه ی مکان روبات دچار خطا می‌شود.

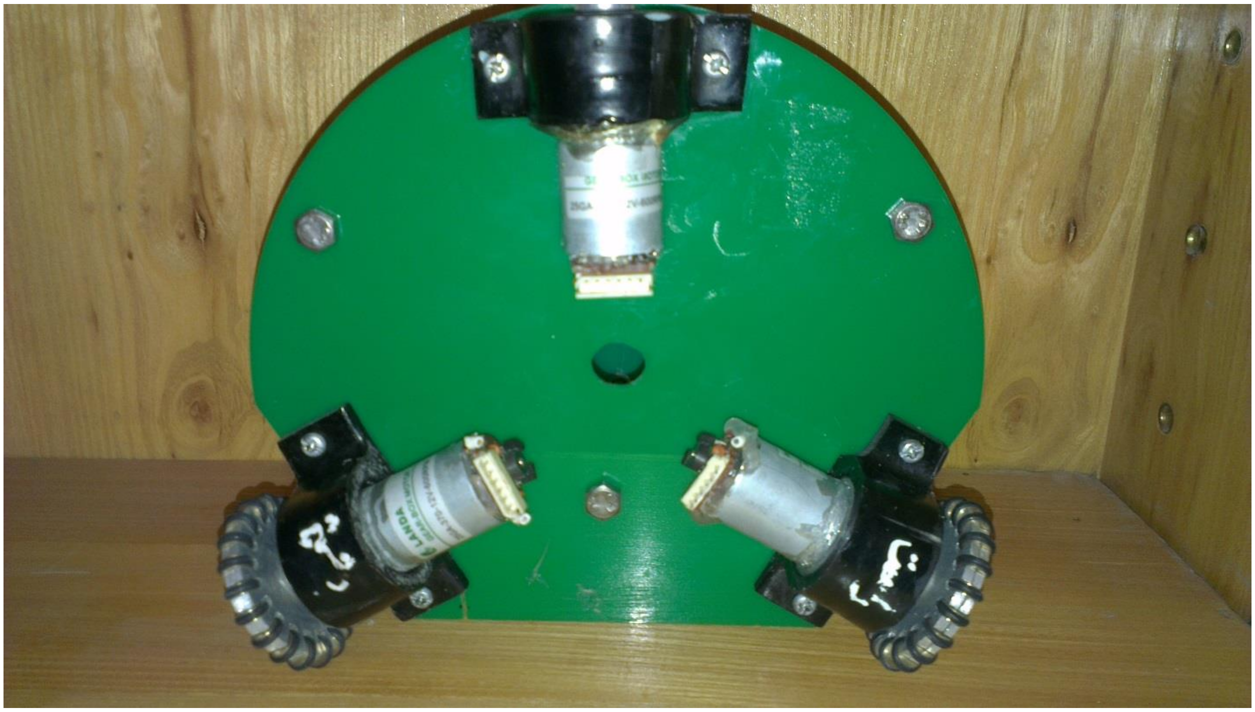
- انکودرهای موتورها باید بسیار دقیق و دارای بازخورد باشند. زیرا همانطور که در پاراگراف قبل گفته شد چنانچه لحظه‌ای خطا در شمارش انکودرها رخ دهد مطابق شکل ۲-۱۵ محاسبه حرکت روبات دچار انحراف می‌شود.
- محاسـبات در دستگاه استوانه‌ای در حالت کلی نسبت به دستگاه دکارتی کمی مشکل‌تر است و بار محاسباتی اندکی بالاتر می‌رود.



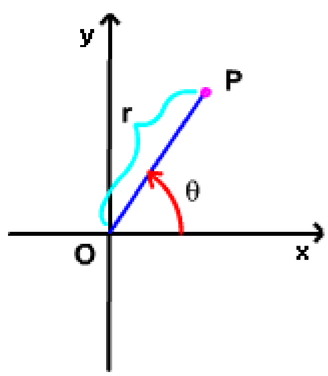
*شکل ۲- ۱۰ نمونه ی ساخته شده ایده شماره ۲ از نمای روبرو، این روبات با سه چرخ خورشیدی که دارای زاویه‌ی ۱۲۰ درجه با یکدیگر هستند مدل می شود. این چرخ‌ها حرکت روبات را در دستگاه مختصات استوانه‌ای مدل سازی می‌کنند.*



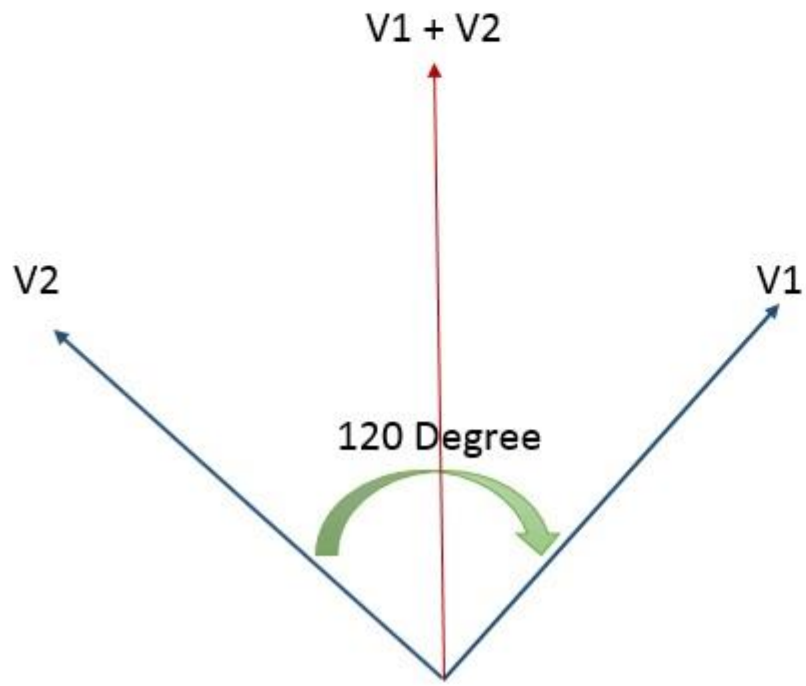
شکل ۲- ۱۱ مکانیک ساخته شده ایده شماره ۲ از نمایی نزدیکتر، چرخ‌های خورشیدی این قابلیت را دارند تا علاوه بر حرکت در حول محور خود، در راستای عمود بر سطح مقطع چرخ نیز حرکت نمایند.



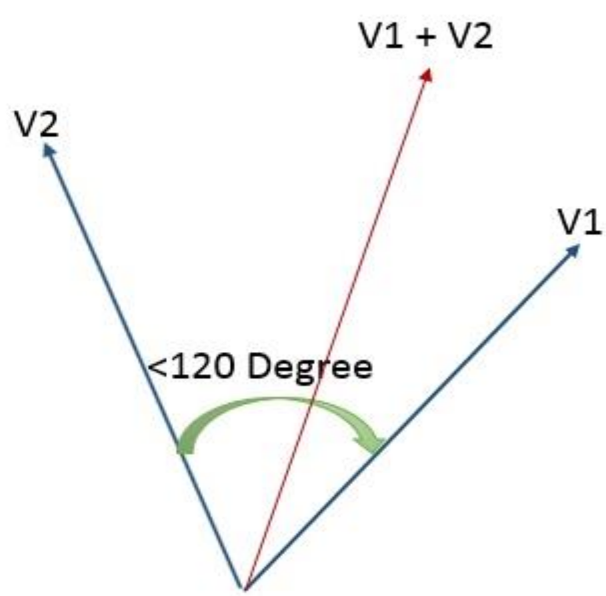
*شکل ۲- ۱۲ نحوه قرارگیری چرخها با زاویه ۱۲۰ درجه نسبت به هم، این زاویه باید به صورت کامل دقیق پیاده‌سازی شود، هرگونه انحراف از این زاویه موجب خطا در پیاده‌سازی محاسبات مکان‌یابی روبات خواهد بود.*




*شکل ۲- ۱۳ دستگاه مختصات استوانه ای، در این دستگاه هر نقطه به وسیله ی یک بردار جابجایی از مرکز و یک زاویه که بردار جابجایی با افق می‌سازد مشخص می‌شود. در اینجا نقطه ی $P$ به وسیله‌ی بردار جابجایی از مرکز r و زاویه $Teta$ مشخص می‌شود.*




شکل ۲- ۱۴ برابر بودن سرعت موتورها و قرارگرفتن بردار سرعت ها با زاویه‌ی ۱۲۰ درجه یکی از الزامات مهم ایده شماره ۲ می‌باشد. در این شکل بردارهای آبی رنگ سرعت موتورها و بردار قرمز رنگ جهت حرکت روبات که برابر با برآیند بردارهای سرعت است،.می باشند.



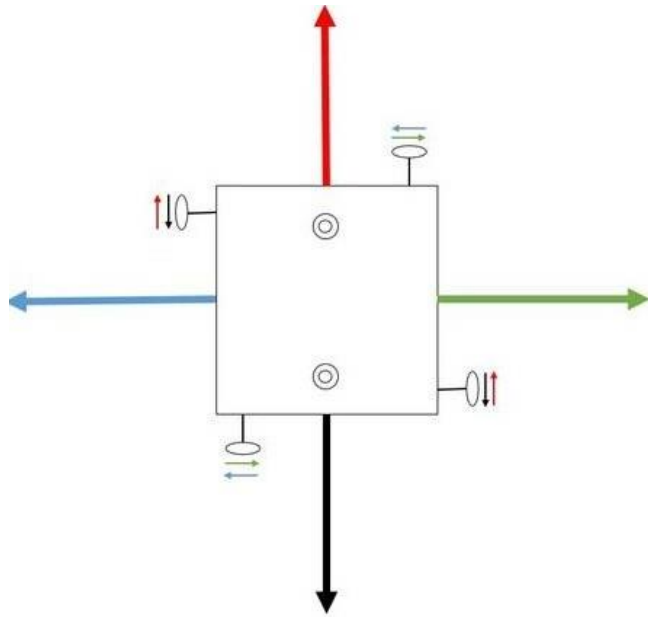

شکل ۲- ۱۵ در شرایطی که به هر دلیل زاویه بین بردارهای سرعت ۱۲۰ درجه نباشد و یا بردارهای سرعت با هم برابر نباشند روبات از مسیر اصلی و موردنظر منحرف می شود.

- بررسی ایده شماره ۳:

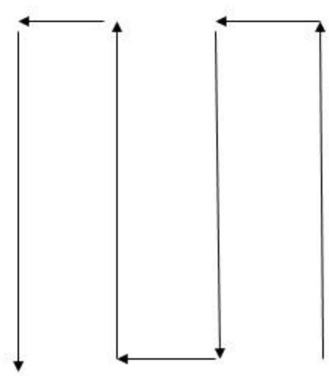
پس از ساخت روبات مطرح شده در ایده شماره ۳ و انجام آزمایش بر روی آن و سنجش کارایی آن، معایب و مزایای آن به شرح زیر مشخص شدند:

مزایا:

- مطابق شکل ۲-۱۶ که نمایی از مکانیک ایده شماره ۳ را از بالا نشان می دهد، این روبات از ۴ چرخ خورشیدی تشکیل شده است. دو چرخ در راستای محور x قرار دارند و دو چرخ آن نیز در راستای محور yها واقع شده‌اند. این موضوع سبب می شود حرکت روبات به جای حرکت در دو بعد به طور همزمان، به حرکت در راستای یکی از دو محور عمودی یا افقی در هر لحظه تبدیل شود. به عبارت دیگر x و y مختصات روبات هیچگاه به طور همزمان تغییر نخواهند کرد. این یک قرارداد است.
- محاسبه‌ی مختصات روبات در هر لحظه بسیار آسان است. زیرا مسافت پیموده شده توسط چرخ‌هایی که در راستای محور y قرار دارند برابر عرض مختصات مکان روبات و مسافت پیموده شده توسط چرخ‌هایی که در راستای محور x قرار دارند برابر طول مختصات مکان روبات می باشد.
- این روبات هیچگاه دور نمی‌زند! (این نیز یک قرارداد است!) حرکت این روبات هنگام تغییر جهت مطابق شکل ۲- ۱۷ خواهد بود و فقط مکان قرار دادی جلو و عقب آن تغییر می‌کند. بنابراین مشکلات دورزدن را که سایر ایده‌ها با آن مواجه بودند نخواهد داشت.
- الگوریتم حرکتی این روبات بسیار ساده است. زیرا در هر لحظه تنها یا در راستای محور x حرکت خواهد داشت و یا در راستای محور y.
- اگر روبات بخواهد از مبداء به مختصات $(x_1,y_1)$ برود کافی است چرخ‌های راستای x آن به اندازه ی $x_1$ و چرخ‌های راستای y آن به اندازه ی $y_1$ جابجایی داشته باشند. این موضوع خطای مکان‌یابی روبات را تا حد بسیار زیادی کاهش می‌دهد.



شکل ۲-۱۶ نمایی از مکانیک ایده شماره ۳ از بالا، چهار دایره‌ی کوچک در چهار سمت روبات معرف چرخهای حرکتی روبات هستند. این چرخها همگی خورشیدی می‌باشند. زاویه‌ای که چرخهای راستای افق با چرخهای راستای قائم می‌سازند ۹۰ درجه است. دایره هایی که در داخل روبات در سمت بالا و پایین قرار دارند، حسگرها و موتورهای گرداننده‌ی حسگر را نشان می‌دهند.



شکل ۲-۱۷ الگوی حرکتی روبات ایده شماره ۳، این روبات هیچگاه دور نخواهد زد، هنگام رسیدن به پایان حرکت در یک جهت، با یک جابجایی در راستای دیگر، مطابق شکل شروع به حرکت در جهت مخالف خواهد کرد. قسمتهای جلو و عقب روبات هیچگاه با یکدیگر عوض نخواهد شد.

معایب:

- به دلیل آنکه روبات هیچگاه نمی‌چرخد باید مطابق شکل ۲-۱۶ حداقل از دو عدد حسگر در قسمت‌های بالا و پایین آن اســتفاده شـود. در صورت استفاده از یک حسگر جمع‌آوری اطلاعات با اندکی خطا انجام خواهد شد. زیرا تصور می شود مساحت روبات به گونه‌ای باشد که حسگر گاهی اوقات روبات را نیز به عنوان مانع تشخیص دهد.
- هزینه ی ساخت روبات به دلیل ساختار مکانیکی پیچیده‌تر آن به نسبت دو ایده‌ی قبلی بالاتر می‌باشد.
- سـاخت روبات، دقت نسـبتا بالایی را می طلبد. زیرا چرخ ها باید کاملا در راسـتای یکدیگر قرار داشته باشند. همچنین چرخ هایی که در یک راسـتا قرار دارند باید بر چرخهایی که در راستای دیگر قرار دارند عمود باشند. البته این دقت در ساخت به مراتب پایین تر از دقت مورد نیاز برای ایده های شماره ۱ و ۲ است.

### ۲-۶-۲ انتخاب آرایش مکانیک نهایی

با توجه به مشـکلات محاسـبه‌ی مختصات روبات و همچنین کاهش یافتن تصاعدی دقت انکودرها، ایده شماره۱ رد می‌شود. همچنین ایده‌ی شماره ۲ نیز به دلیل مشکل بودن کنترل حرکت روبات و محاسبه‌ی مختصات روبات به وسیله‌ی انکودرها و همچنین به دلیل پیچیدگی محاسبات در دستگاه مختصات استوانه‌ای نیز رد شد.

ایده‌ی شماره‌ی ۳ دارای ۴ چرخ خورشیدی بوده و حرکت روبات را بجای ۲ بعد صرفا در یک زمان واحد، در راستای یکی از دو محور x یا y محدود کرده است. همچنین محاسبه مختصات روبات در این ایده ساده‌تر است و از دقت بسیار بالاتری برخوردار است. با توجه به این مزایا و با صرف نظر از معایب کم آن این طرح ساخته شد و به عنوان طرح اصلی روبات سکوی مورد نظر انتخاب گردید.

### ۲-۶-۳ بررسی مواد مطرح شده برای جنس شاسی

پس از طرح و بررسی دقیق ایده‌های مختلف در مورد جنس شاسی بدنه که در بخش ۲-۲ بیان شدند، نقاط ضعف و قوت هر ایده به صورت زیر مشخص شد:

- پلکسی گلاس:

مزایا:

- وزن مناسب
- ظرافت بالا
- سازگاری با فناوری‌های برش موجود در کشور
- امکان پیاده‌سازی دقیق طراحی
- در دسترس بودن با ابعاد دلخواه

معایب:

- شکننده بودن
- قیمت نسبتا بالا در مقایسه با سایر مواد
- توانایی تحمل جرم نسبتا پایین‌تر نسبت به سایر مواد

- آلومینیوم:

مزایا:

- انعطاف پذیری نسبتا بالاتر نسبت به سایر مواد
- تحمل وزن بالا
- در دسترس بودن با ابعاد دلخواه
- پایین‌ترین قیمت ممکن در میان سایر مواد

معایب:

- انعطاف‌پذیری نسبتا پایین در ضخامت کم
- وزن نسبتا بالاتر در ضخامت‌های زیاد
- مزاحمت برای سخت افزار روبات به صورت افت پتانسیل، ایجاد نویز، اتصال کوتاه در اثر رسانایی
- عدم امکان پیاده‌سازی دقیق طراحی به دلیل عدم وجود واسط میان رایانه و دستگاه های برش آلومینیوم

- تفلون:

مزایا:

- انعطاف‌پذیری نسبتا خوب (بیشتر از پلکسی گلاس و کمتر از آلومینیوم)
- وزن نسبتا کمتر از آلومینیوم
- توان تحمل ضربه ی نسبتا بالاتر از دو ماده دیگر
- توان تحمل حرارت نسبتا بالا از دو ماده دیگر

معایب:

- عدم امکان پیاده سازی دقیق طراحی به دلیل عدم وجود واسط میان رایانه و دستگاه های برش تفلون
- در دسترس نبودن در ابعاد مختلف در بازار به دلیل آنکه تفلون فقط به صورت سیلندر فروخته می‌شود. درنتیجه محدودیت‌ها و مشکلات بسیار زیاد در ساخت

### ۲-۶-۴ انتخاب ماده‌ی تشکیل دهنده‌ی شاسی

با بررسی نقاط ضعف و قوت ایده‌های مطرح شده در بخش ۲-۵-۳ در مورد جنس شاسی بدنه، این نتیجه حاصل شد که استفاده از پلکسی‌گلاس به دلایل زیر بهتر از سایر ایده‌ها می‌باشد و این ماده با نیازها و شرایط روبات Platform تطابق بیشتری دارد:

- وزن کمتر از سایر مواد
- سازگاری با فناوری‌های برش موجود در کشور
- امکان پیاده سازی دقیق طراحی
- در دسترس بودن با ابعاد دلخواه

در نتیجه پلکسی‌گلاس به عنوان ماده‌ی مناسب برای ساخت شاسی روبات مورد استفاده قرار گرفت.

### ۲-۶-۵ بررسی موتورهای حرکتی پیشنهادی

- موتور DC:

مزایا:

- راه اندازی بسیار آسان در میان سایر موتورها
- نصب مکانیکی آسانتر از موتور پله‌ای
- صرفه جویی در تعداد خطوط گذرگاه زیرا موتور DC اغلب با دو سیم راه اندازی می‌شود
- در دسترس بودن
- قیمت مناسب نسبت به سایر موتورها
- طول عمر نسبتا بالا در مقایسه با سایر موتورها
- پوشش گستره وسیعی از سرعت‌ها و گشتاورها
- سازگاری شافت موتور با بسیاری از چرخ‌های استاندارد

معایب:

- کنترل نسبتا دشوارتر از موتور پله‌ای
- نیاز به پیاده سازی مدار واسط برای کنترل
- نیاز به نصب شمارشگر بر روی موتور برای کنترل آن
- امکان کنترل سرعت در بازه‌ای به نسبت محدودتر از موتور پله‌ای

- موتور پله ای:

مزایا:

- عدم نیاز به نصب شمارشگر صرفا برای کنترل آن
- کنترل بسیار ساده‌تر از موتور DC
- قابلیت تنظیم سخت‌افزاری دقت موتور
- پوشش گستره ی نسبتا وسیعی از دقت ها و گشتاورها
- امکان تنظیم سرعت

معایب:

- قیمت به نسبت بالاتر از موتورهای DC
- اشغال کردن حداقل ۴ خط گذرگاه ورودی/خروجی برای راه‌اندازی
- نویزپذیری بسیار بالای خطوط راه‌اندازی
- تولید نویز بسیار بالاتر از موتور DC به دلیل سوئیچ کردن‌های مداوم خطوط کنترلی
- نصب مکانیکی نسبتا دشوارتر از موتور DC
- عدم سازگاری شافت موتور با بسیاری از چرخ های استاندارد و در نتیجه نیاز به ایجاد تغییرات مکانیکی

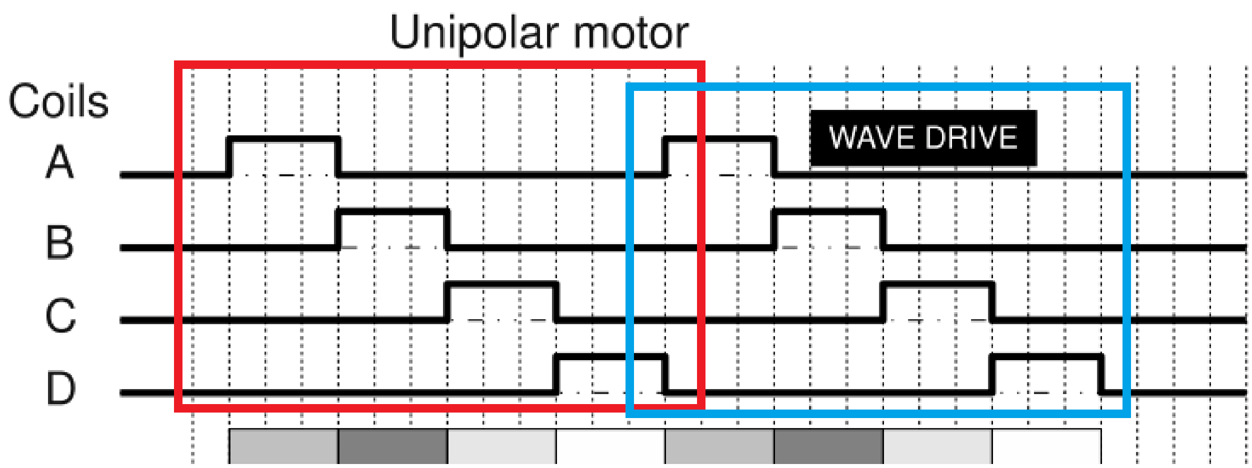
### ۲-۶-۶ انتخاب موتور حرکتی مناسب

با توجه به معایب و مزایای مشخص شده برای هر یک از موتورها و مقایسه‌ی آنها با معیارهای مورد نظر در طراحی روبات سکو این نتیجه حاصل شد که استفاده از موتور DC به دلایل زیر انتخاب به مراتب مناسب تری برای موتور حرکتی روبات سکو می باشد:

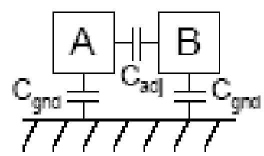
- قیمت مناسب‌تر
- نصب مکانیکی آسان‌تر
- سازگاری با بسیاری از چرخ های موجود در بازار روباتیک
- عمر عملیاتی نسبتا بالاتر

مزیت دیگری که باعث انتخاب موتور DC می شود، تولید نویز پایین آن در مقابل موتور پله‌ای است. به طور متوسط برای هر ۴ تا ۶ حرکت پله در موتور پله‌ای باید به طور تناوبی یکی از خطوط کنترلی موتور پله‌ای به سطح بالا برود و بقیه در سطح پایین بمانند. به عبارتی دیگر به ازای هر یک پله حرکت در موتور پله ای یک عمل سوئیچینگ در خطوط کنترلی خواهیم داشت. شکل ۲-۱۸ سیگنال‌های خطوط کنترلی یک موتور پله‌ای که دارای گذرگاه کنترلی ۴ کاناله می‌باشد را نشان می دهد. همانطور که در این شکل ملاحظه می شود، هر یک از سیگنال‌های کنترلی در هر ۴ پله حرکت یک بار به مدتی مشخص در سطح ولتاژ بالا قرار خواهد گرفت و سپس به سطح ولتاژ صفر خواهد رفت. مدت زمان بالا بودن سیگنال متناسب با سرعت موتور می‌باشد. بدیهی است که هر چقدر سرعت موتور بیشتر باشد، مدت زمان یک بودن کمتر است. در شکل ۲-۱۸ مستطیل قرمز رنگ نشان دهنده ی سیگنال‌های کنترلی برای حرکت ۴ پله‌ی موتور و مستطیل آبی رنگ نشان دهنده‌ی تکرار سیگنال‌های کنترلی برای حرکت دوباره‌ی همان ۴ پله است. این تغییر سطح ولتاژ و سوئیچینگ موجب ایجاد ظرفیت‌های خازنی پویا میان خطوط گذرگاه در مدار روبات می شود. این ظرفیت‌های خازنی در مقابل تغییر سطح ولتاژ خطوط کنترلی که توسط پردازنده‌ها فرمان داده می شود مقاومت می‌کنند و باعث می شوند ولتاژ به سطح مورد نظر نرسد. شکل ۲-۱۹ اثرات خازنی خطوط گذرگاه را

نسبت به یکدیگر و همچنین نسبت به زمین نشان می‌دهد. همانطور که در این شکل ملاحظه می شود، هر دو سیم مجاور، علاوه بر اثر خازنی با زمین مدار که به آن $C_{gnd}$ گفته می‌شود، دارای اثر خازنی متقابل با یکدیگر می‌باشند که به آن به اختصار خازن مجاورت یا $C_{adj}$ گفته می‌شود. در جدول موجود در شکل ۲-۱۹ مقادیر مختلف خازن مجاورت به با توجه به شرایط مختلف سیم B بیان شده است. این شرایط به ترتیب شامل زمانی که ولتاژ B ثابت است، زمانی که ولتاژ B مشابه با A تغییر می‌کند و زمانی که ولتاژ B در خلاف جهت ولتاژ A تغییر می‌کند مشخص شده است.[۱۸]




شکل ۲- ۱۸ دیاگرام سیگنال های کنترلی موتور پله ای برای حرکت ۸ پله متوالی, هر یک از مستطیل های رنگی یکی از ۴ حرکت متوالی را مدل می کنند.




| B | ΔV | $C_{eff(A)}$ | MCF |
|---|---|---|---|
| Constant | $V_{DD}$ | $C_{gnd} + C_{adj}$ | 1 |
| Switching with A | 0 | $C_{gnd}$ | 0 |
| Switching opposite A | $2V_{DD}$ | $C_{gnd} + 2\ C_{adj}$ | 2 |

شکل ۲- ۱۹ خازن های مجاورت میان دو سیم در گذرگاه: هرگاه دو سیم مجاور یکدیگر در یک گذرگاه عمل سوئیچینگ انجام دهند، بسته به جهت سوئیچینگ یک ظرفیت خازنی میان آنها ایجاد خواهد شد.

معادله شماره ۲-۱ تغییر ولتاژ در اثر سوئیچینگ یک سیم در سیم دیگر زمانی که سیمی که تحت تاثیر قرار می‌گیرد، بدون ولتاژ است را نشان می دهد:

$$\Delta Vvictim = \frac{Cadj}{Cgnd-v+Cadj}\,\Delta Vagressor$$ معادله ۲-۱

در معادله ۲-۱ $\Delta Vvictim$ تغییر ولتاژ سیمی است که تحت تاثیر قرار می‌گیرد، $\Delta Vagressor$ تغییر ولتاژ سیمی است که عمل سوئیچ را انجام می‌دهد (از صفر به یک یا از یک به صفر)، Cadj ظرفیت خازنی مجاورت بین دو سیم است و Cgnd-v ظرفیت خازنی سیمی که تحت تاثیر قرار می‌گیرد نسبت به زمین مدار می‌باشد.

همچنین معادله شماره ۲-۲ تغییر ولتاژ در اثر سوئیچینگ یک سیم در سیم دیگر زمانی که سیمی که تحث تاثیر قرار می‌گیرد دارای ولتاژ است را نشان می‌دهد:

$$\Delta Vvictim = \frac{Cadj}{Cgnd-v+Cadj}\frac{1}{1+k}\,\Delta Vagressor$$ معادله ۲-۲

همانند معادله ۲-۱ در معادله ۲-۲ $\Delta Vvictim$ تغییر ولتاژ سیمی است که تحت تاثیر قرار می گیرد، $\Delta Vagressor$ تغییر ولتاژ سیمی است که عمل سوئیچ را انجام می‌دهد (از صفر به یک یا از یک به صفر)، Cadj ظرفیت خازنی مجاورت بین دو سیم است و Cgnd-v ظرفیت خازنی سیمی که تحت تاثیر قرار می‌گیرد نسبت به زمین مدار می‌باشد. مقدار k یک ضریب ثابت است که از معادله۲-۳ بدست می‌آید.

$$k = \frac{\tau agressor}{\tau victiom} = \frac{Ragressor(Cgnd-a+Cadj)}{Rvictim(Cgnd-v+Cadj)}$$ معادله ۲-۳

در معادله ۲-۳ $\tau victiom$ ثابت زمانی سیمی است که تحت تاثیر قرار می‌گیرد. $\tau agressor$ ثابت زمانی سیمی است که عمل سوئیچ را انجام می دهد. Ragressor مقاومت سیمی است که سوئیچ می‌کند و Rvictim مقاومت سیمی است که تحت تاثیر قرار می‌گیرد. همچنین Cgnd-a ظرفیت خازنی نسبت به زمین سیم سوئیچ کننده است. سایر نمادها در معادله۲-۲ شرح داده شد.

به این پدیده که سوئیچ کردن یک سیم در یک گذرگاه باعث تغییر سطح ولتاژ در سیم‌هایی می شود که سوئیچ نمی‌کنند و سطح ولتاژ آنها ثابت است، نویز هم‌شنوایی یا Cross Talk Noise گفته می‌شود. [۱۸]

## ۲-۶-۷ بررسی موتورهای گرداننده‌ی حسگر

همانطور که در بخش ۲-۳-۲ ملاحظه می‌شود، دو نوع موتور حرکتی برای گردش حسگر(ها) پیشنهاد شدند: سرو موتور و موتور پله‌ای. همچنین موتور DC به دلیل شـــرایط ســـخت کنترل آن و دقت به مراتب پایین‌تر آن نســـبت به موتورهای ســـرو و پله‌ای در همان مرحله‌ی ایده‌پردازی حذف شـد. در ادامه به بررسـی مزایا و معایب موتورهای پله‌ای و سرو برای انجام ماموریت گردش حسگر می‌پردازیم. لازم به توجه اسـت که گرچه در بخش ۲-۶-۶ نیز به بررسـی مزایا و معایب موتور پله‌ای پرداخته شـد، اما در آن بخش بیشتر تمرکز بر روی توانایی این نوع موتورها در به حرکت در آوردن روبات بود. در مقابل در این بخش از نقطه نظر قابلیت آنها در کنترل گردش حسگر به آن می‌پردازیم.

- موتور سرو:

مزایا:

- سرعت واکنش بسیار سریعتر از سایر موتورها
- حرکت مستقیم به یک زاویه (هنگامی که می‌خواهد به یک زاویه مشخص برود، تمام زاویه‌های قبلی و کوچکتر از آن را با سـرعت بسیار زیاد طی می‌کند و نیاز نیست به تک تک زاویه‌ها برود و در آنها توقف داشته باشد به عبارتی دیگر تغییر زاویه بسیار سریع دارد)
- دارای زاویه صفر مشخص و ثابت
- قابلیت بازگشت به زاویه صفر
- دقت بسیار بالاتر از سایر موتورها
- کنترل بسیار راحت توسط پردازنده
- امکان اتصال مستقیم به پردازنده بدون هیچ گونه مدار واسط و درایور
- صرفه جویی بسیار بالا در خطوط گذرگاه (این موتور فقط یک سیم کنترلی برای دریافت "پالس" نیاز دارد)
- سازگاری با سطوح ولتاژ دیجیتال برای تغذیه‌ی موتور و عدم نیاز به سطوح بالاتر ولتاژ
- پشتیبانی از گشتاورهای بسیار بالا در ابعاد نسبتا کوچک موتور
- عدم نیاز به انکودر برای دریافت موقعیت مطلق موتور
- نگهداری موقعیت در اثر اعمال گشتاور از خارج از روبات

معایب:

- ایجاد نویز هم‌شنوایی بر روی خطوط مجاور
- تاثیر پذیری از نویز هم شنوایی و ظرفیت‌های خازنی مجاور و در نتیجه خارج شدن کنترل از فرمان پردازنده
- مصرف توان استاتیک برای ماندن در یک زاویه مشخص
- مصرف توان و جریان بسیار زیاد در هنگام تغییر زاویه (تا ۱۰ برابر حد معمول به هنگام ماندن در یک زاویه)
- قیمت نسبتا بالاتر در مقایسه با سایر موتورها
- محدودیت در کنترل تعداد سـرو ها توسـط پردازنده به صـورت مسـتقیم ( تعداد سرو قابل کنترل توسط یک پردازنده با تعداد کانال‌های تایمر آن متناسب است)
- نیاز به واسط شافت برای اتصال به سایر اجزا
- پشـتیبانی از گسـتره ی ۹۰- درجه تا ۹۰+ درجه (۱۸۰ درجه کامل) و عدم پشـتیبانی گسـتره ۳۶۰ درجه از محیط اطراف

- حساسیت به فرکانس مشخص پالس کنترلی و عدم کارایی کامل در سایر فرکانس‌ها
- نیاز به پیکربندی خاص بر روی پردازنده برای تولید پالس

- موتور پله ای:

مزایا:

- راه اندازی نسبتا آسانتر از موتور سرو
- پوشش ۳۶۰ درجه محیط
- مصرف توان به نسبت پایین تر از سرو
- عدم مصرف توان استاتیک برای ماندن در یک زاویه
- عدم محدودیت در تعداد کنترل توسط پردازنده
- عدم نیاز به پیکربندی‌های خاص بر روی پردازنده
- عدم نیاز به واسط شافت برای اتصال به اجزا
- عدم مصرف توان بالا به هنگام تغییر زاویه
- عدم وابستگی به فرکانس
- امکان تنظیم سرعت

معایب:

- نیاز به نصب انکودر برای تشخیص درست زاویه
- دقت پایین در مقایسه با موتور سرو
- عدم داشتن صفر مطلق و ثابت و مشخص
- عدم داشتن قابلیت بازگشت به صفر به طور مستقیم از هر زاویه دلخواه
- افزایش ابعاد موتور با افزایش گشتاور
- عدم نگهداری موقعیت در اثر اعمال گشتاور از خارج از روبات
- عدم پشتیبانی از حرکت مستقیم به یک زاویه و پیمایش ترتیبی زوایا حرکت مستقیم به یک زاویه (هنگامی که می‌خواهد به یک زاویه مشخص برود، تمام زاویه های قبلی و کوچکتر از آن را با سرعت کاملا یکنواخت طی می‌کند و به تک تک زاویه‌ها می رود و در آنها توقف داشته دارد به عبارتی دیگر تغییر زاویه بسیار آهسته تر از سرو دارد)

### ۲-۶-۸ انتخاب موتورهای گرداننده‌ی حسگر

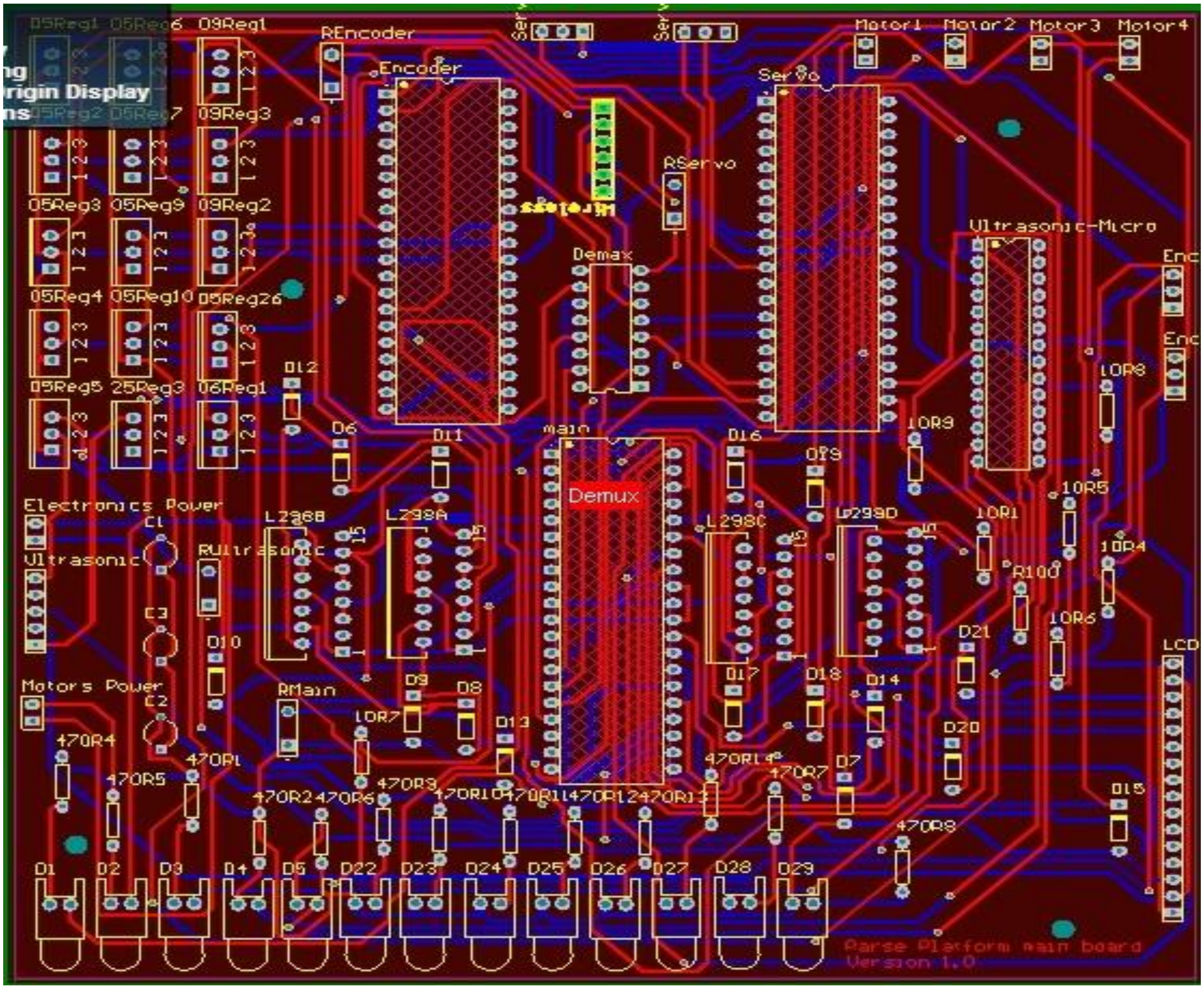
با توجه به آنکه گردش حسگر با هدف ایجاد یک تصویر دقیقَ شفاف و پیوسته از محیط اطراف می‌باشد، بدیهی است عواملی مانند دقت بالا در حرکت، دقت بالا در مکان‌یابی و دقت بالا در شروع مجدد نقش بسیار مهمی در اجرای صحیح عملیات موتور گرداننده‌ی حسگر خواهد داشت. با توجه به این عوامل و با در نظر گرفتن مزایا و معایب نوع موتورهای انتخاب شده برای حسگر که در بخش۲-۶-۷ مطرح شد، این نتیجه حاصل شد که موتور سرو به دلیل ارضا کردن شرایط لازم برای موتور گرداننده ی حسگر انتخاب به مراتب مناسب‌تری می‌باشد. البته در بخش ۲-۶-۷ مزایا و معایب موتورهای گرداننده‌ی حسگر روبات از نقطه نظرهای متنوع بحث شدند و همانطور که ملاحظه شد موتور پله‌ای در بعضی بخش‌ها مانند مصرف توان ایستا و پویا نسبت به موتور سرو دارای برتری می‌باشد. اما همانطور که در ابتدای این قسمت گفته شد، در انتخاب موتور گرداننده‌ی حسگر، در درجه ی اول دقت مهم‌ترین عامل است. به وضوح دقت مهمترین ویژگی موتورهای سرو می‌باشد. بنابراین گرچه موتور سرو در برخی جنبه ها برتر از موتور پله‌ای نخواهد بود، اما به دلیل ارضاء کردن شرایط اساسی موتور گرداننده‌ی حسگر، انتخاب موجه‌ای می باشد.

همانطور که در بخش ۲-۶-۷ ذکر شد، یکی از معایب موتورهای سرو عدم پوشش دهی ۳۶۰ درجه‌ای می‌باشد. به عبارتی دیگر، موتورهای سرو قادر نخواهند بود تمامی ۳۶۰ درجه‌ی اطراف خود را پویش نمایند. موتورهای سرو تنها می توانند محدوده‌ی زاویه‌ای ۱۸۰ درجه را با شروع از زاویه ی ۹۰- درجه و خاتمه به زاویه ی ۹۰+ درجه پویش نمایند.[۹] این مسئله باعث می‌شود تا برای ایجاد یک تصویر پیوسته از محیط، حداقل به دو عدد موتور سرو نیاز باشد.

### ۲-۵-۹ بررسی و جمع‌بندی ایده‌های مطرح شده در بخش پردازش سخت‌افزاری

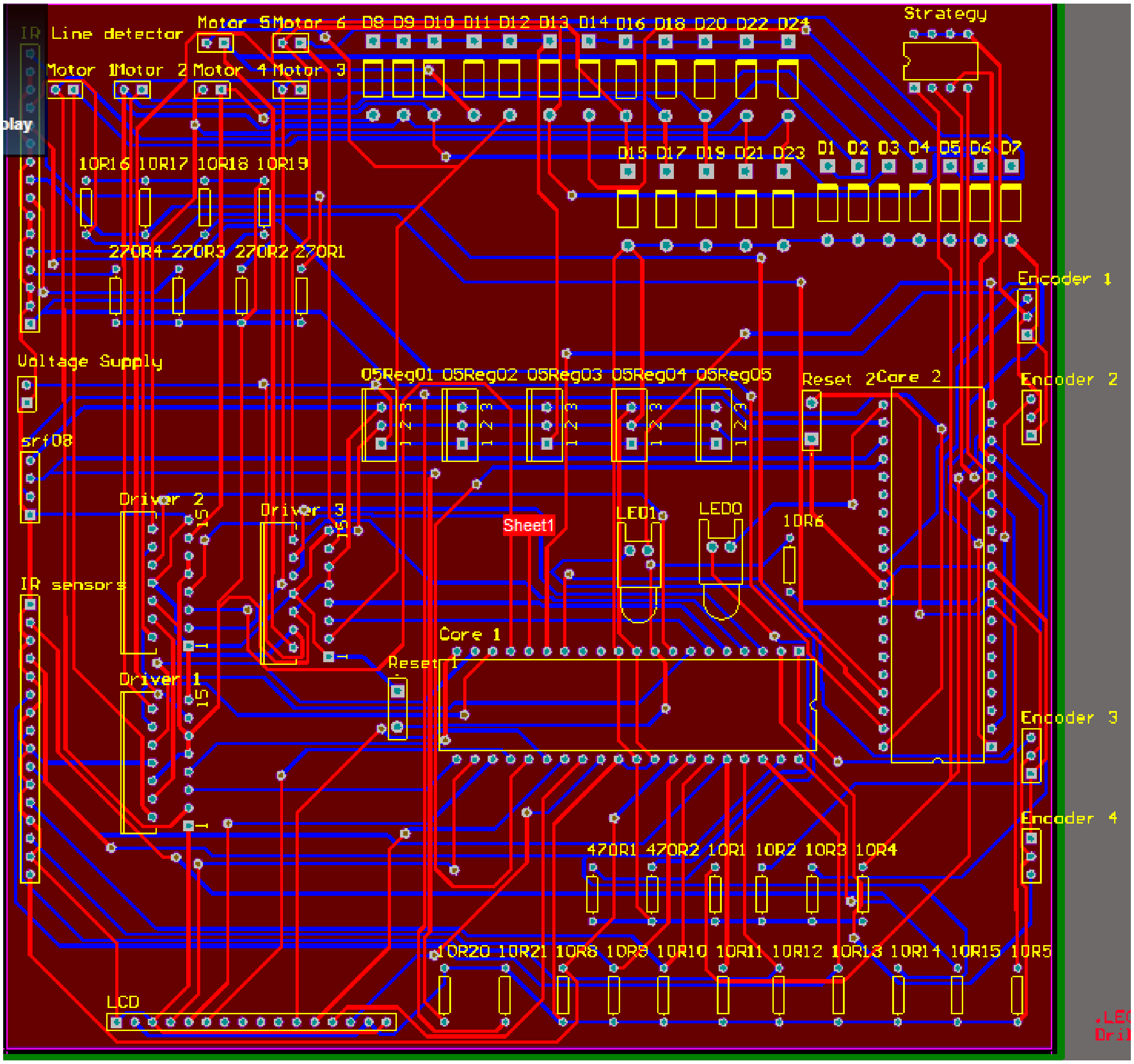
با پیاده سازی تجربی هر دو نوع پردازش توزیع شده و متمرکز در طراحی مدار روبات سکو، این نتیجه حاصل شد که شیوه‌ی پردازشی مجزا برای پردازش سخت‌افزاری اطلاعات مناسب‌تر است. زیرا با پیاده‌سازی این شیوه و توزیع بار پردازشی بر روی چند پردازنده به صورت کاملا مستقل، سرعت پردازش روبات افزایش می‌یابد. همچنین به دلیل مستقل بودن برنامه‌ی بخش‌های مختلف از یکدیگر، برنامه‌ریزی، عیب‌یابی برنامه و ارتقاء برنامه‌ی بخش‌های مختلف روبات به آسانی انجام می‌پذیرد. این در حالی است که در پیاده‌سازی پردازش به صورت متمرکز، به دلیل تمرکز بار پردازشی بر روی حداقل تعداد پردازنده‌ها، سرعت پردازش به مراتب پایین‌تر از پردازش توزیع شده است. همچنین در پردازش متمرکز، عیب‌یابی برنامه، ارتقاء آن و برنامه‌ریزی روبات به دلیل یک پارچه بودن برنامه و در نتیجه حجم زیاد آن، به سهولت پردازش توزیع شده نخواهد بود.

شکل ۲-۲۰ صورت چاپی نمونه‌ی اولیه‌ی مدار اصلی (Motherboard ) روبات را نشان می‌دهد. هدف از قرار دادن این شکل در این قسمت، فقط نشان‌دادن طراحی سیستم پردازشی روبات به صورت سیستم پردازشی مجزا می‌باشد و جزئیات پیاده‌سازی آن به صورت کامل به همراه نمونه‌ی نهایی مدار روبات در فصل بعدی بیان می‌شود. همانطور که در شکل ۲-۲۰ مشاهده می‌شود، بار پردازشی روبات بر روی ۴ عدد پردازنده توزیع شده است. پردازنده‌ی Ultrasonic وظیفه‌ی پردازش اطلاعات حسگر فراصوت را برعهده دارد. پردازنده‌ی Servo کنترل موتور سرو را بدست می‌گیرد. پردازنده ی Encoder بار پردازشی انکودرهای چرخ های روبات را بر عهده دارد و نهایتا پردازنده‌ی Main کنترل روبات را بر عهده دارد و با پردازنده‌های دیگر در تعامل است.




*شکل ۲- ۲۰ صورت چاپی نمونه ی اولیه مدار اصلی روبات، این بورد از ۴ عدد پردازنده‌ی AVR تشکیل شده است. سیستم پردازشی این برد مجزا می‌باشد. همچنین ابعاد این برد به سبب سیستم پردازش توزیع شده ۲۰ سانتی‌متر در ۲۰ سانتی‌متر است.*

شکل ۲-۲۱ صورت چاپی مدار اصلی روبات با پردازش متمرکز را نشان می دهد. همانطور که ملاحظه می شود شکل‌های ۲-۲۰ و ۲-۲۱ در جزئیات زیادی با یکدیگر متفاوت هستند. دلیل اصلی این موضوع آن است که هدف از طراحی مدار شکل ۲-۲۱، سنجش سرعت و کارایی پردازش متمرکز در پیاده سازی روبات سکو می‌باشد. قطعات و جزئیات بیشتری که در شکل ۲-۲۰ ملاحظه می‌شود، پس از تصویب پردازش توزیع شده و با هدف پیاده سازی سخت افزاری مدار اصلی روبات اضافه شدند. این در حالی است که همانطور که گفته شد مدار شکل ۲-۲۱ فقط با هدف آزمایشی طراحی گردید. در سنجش هردو مدار ۲-۲۰ و ۲-۲۱ با یک برنامه ی یکسان، با هدف استانداردسازی و یکسان‌سازی هر دو مدار، قطعات و بخش های غیر یکسان آنها از مدار خارج شدند. با اندازه گیری زمان اجرای برنامه و تکرار آن به دفعات، این نتیجه حاصل شد که پردازش توزیع شده بین ۲ تا ۲,۴ برابر سریعتر از پردازش متمرکز می باشد.




*شکل ۲- ۲۱ صورت چاپی مدار آزمایش پردازش متمرکز، این بورد فقط از دو عدد پردازنده AVR تشکیل شده است. مساحت این برد به سبب پیاده‌سازی پردازش متمرکز از برد شکل ۲-۲۰ کمتر است. ابعاد این برد چاپی ۱۵ سانتی‌متر در ۱۵ سانتی‌متر می‌باشد.*

### ۲-۶-۱۰ بررسی طراحی ماژولار و غیرماژولار مدار

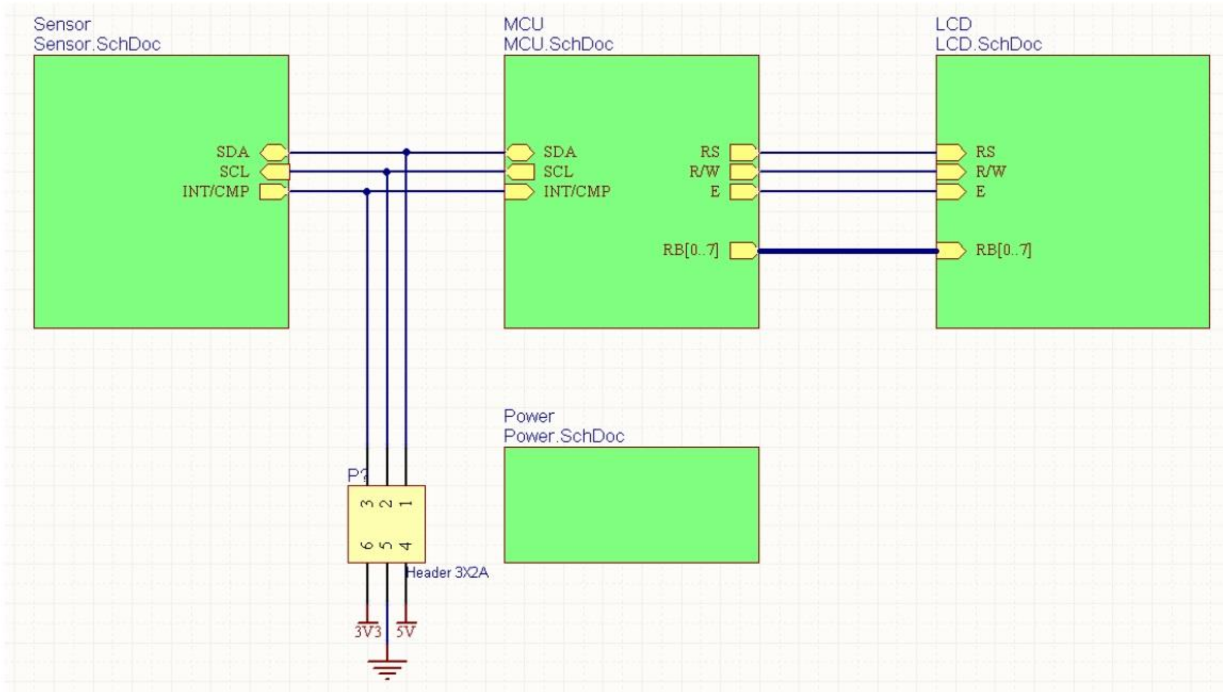
با پیاده سـازی مدار به صورت ماژولار و غیر ماژولار و مطالعه و بررسی کارایی آنها مزایا و معایب طراحی ماژولار و غیر ماژولار به شرح زیر مشخص شدند:[۱۹]

- طراحی ماژولار:

شکل شماره ۲-۲۲ نمونه‌ی شماتیک مدار طراحی شده به صورت ماژولار را نشان می دهد. با توجه به این شکل می‌توان نکات زیر را در مقایسه با طراحی غیرماژولار استخراج کرد:

- مزایا:
  - عیب‌یابی ساده
  - وجود مرز دقیق و شفاف میان بخش های مختلف مدار
  - طراحی ساده
  - سهولت بررسی طراحی
  - سهولت ارتقاء هر بخش مدار
  - سهولت تغییر مدار
  - سهولت تغییر کاربری کل پروژه
  - امکان استفاده از هر بخش در طراحی ها و پروژه های آینده

- معایب:
  - نیاز به فضای بیشتر برای طراحی مدار
  - افزایش مساحت مدار
  - افزایش توان مدار
  - هزینه‌ی نسبتا بالای پیاده‌سازی مدار به دلیل آنکه هر بخش سیستم راه‌اندازی و ورودی و خروجی خاص خود را داراست
  - نیازمند زمان به مراتب بالا برای طراحی مدار



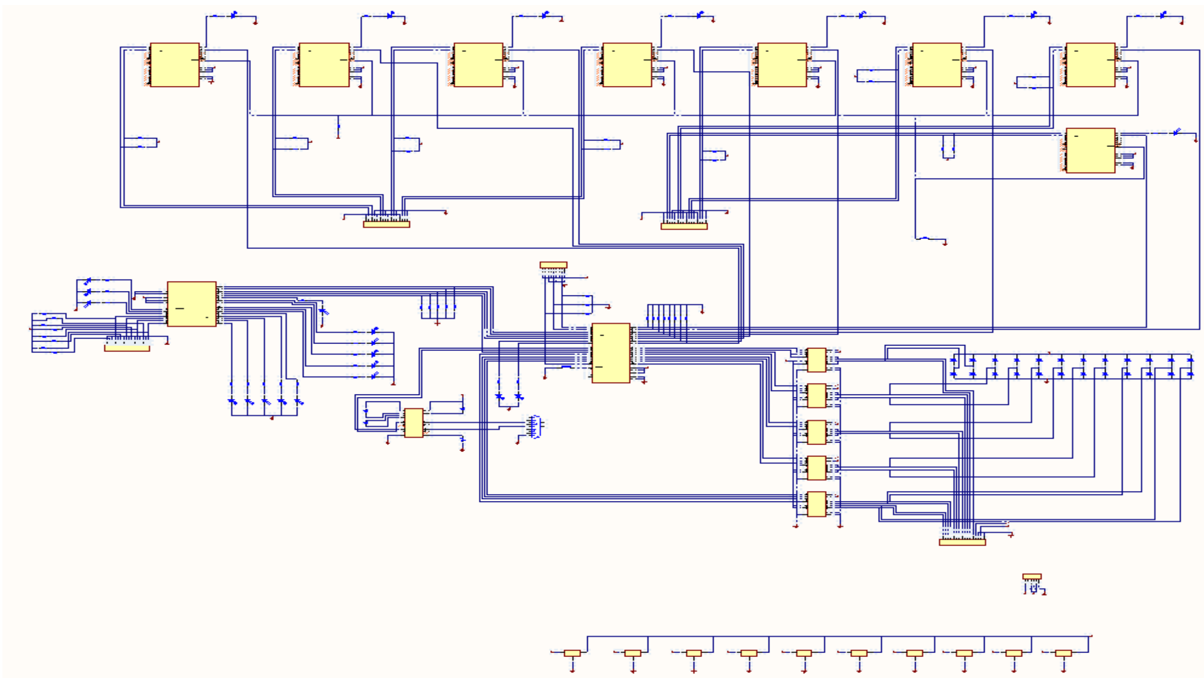

*شکل ۲- ۲۲ یک نمونه طراحی مدار به صورت ماژولار، در این طراحی مدار به بخش‌هایی کاملا مستقل(کادرهای سبز رنگ) تقسیم‌بندی می‌شود. مرز شفاف و دقیقی میان این بخش‌ها وجود دارد. این بخش‌ها فقط از طریق درگاه‌های خود*(اشکال زرد رنگ درون کادرها) با سایر بخش‌ها در ارتباط هستند.

- طراحی غیر ماژولار:

شکل شماره ۲-۲۳ یک نمونه شماتیک مدار غیرماژولار را نشان می‌دهد. این مدار تقریبا همان ساختار، قطعات و کارایی مدار شکل ۲-۲۲ را دارد. با وجود اینکه این دو مدار در نحوه‌ی سخت افزاری پردازش با یکدیگر مشترک هستند و هردو پردازش توزیع شده دارند، در طراحی ماژولار یا غیر ماژولار با یکدیگر متفاوت هستند. با توجه به شکل ۲-۲۳ نکات زیر در مورد طراحی غیرماژولار در مقایسه با طراحی ماژولار استخراج می‌شوند:

- مزایا:
  - هزینه طراحی پایین
  - کاهش توان و مساحت مدار
  - هزینه پیاده‌سازی پایین به دلیل اشتراک بسیاری از قطعات در میان بخش‌های مختلف
  - صرفه جویی در فضای طراحی، نگهداری و پیاده‌سازی
- معایب:
  - عدم وجود مرز دقیق و شفاف میان بخش‌های مختلف مدار
  - دشواری عیب‌یابی
  - دشواری طراحی
  - دشواری بررسی طراحی
  - عدم امکان ارتقاء هر بخش مدار به صورت مجزا
  - دشواری بسیار تغییر مدار

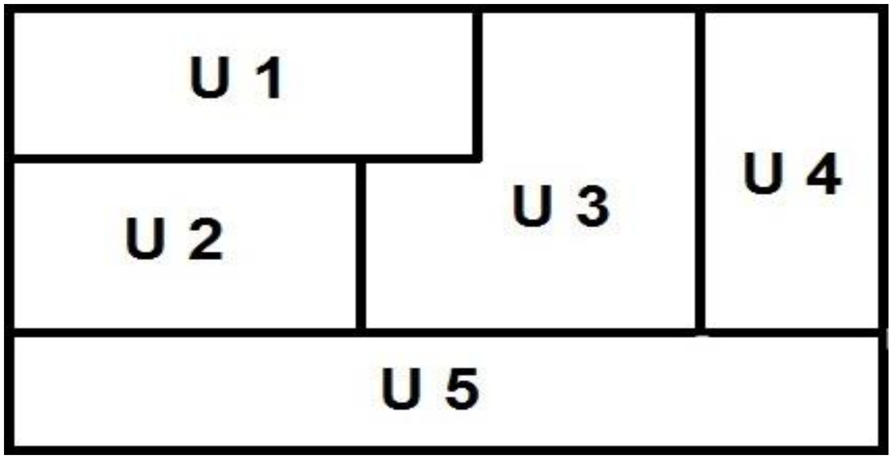
- عدم امکان تغییر کاربری کل پروژه
- عدم امکان امکان استفاده از هر بخش در طراحی‌ها و پروژه‌های آینده



*شکل ۲-۲۳ نمونه شماتیک طراحی شده به صورت غیر ماژولار، همانطور که ملاحظه می‌شود، هیچگونه مرز دقیق و قابل تفکیک میان اجزای مدار وجود ندارد. در این حالت نمی‌توان به سهولت مدار را به بخش‌هایی کاملا مجزا تقسیم‌بندی کرد.*

### ۲-۶-۱۱ جمع بندی نکات طراحی ماژولار و غیرماژولار

ماهیت سکو بودن پروژه ایجاب می‌کند که مدار اصلی روبات انعطاف‌پذیر و قابل ارتقاء باشد. همچنین به سادگی تغییر پذیرد و عیب‌یابی آن به سهولت انجام شود. با درنظر گرفتن این شرایط و مقایسه‌ی آن با ویژگی‌های مطرح شده در بخش ۲-۶-۱۰، این نتیجه حاصل می‌شود که طراحی ماژولار برای مدار روبات سکو انتخاب مناسبی می‌باشد. در طراحی ماژولار مطابق شکل شماره ۲-۲۴، مدار به بخش‌هایی کاملا مستقل به نام واحد (Unit) تقسیم بندی می‌شود. هر واحد کاملا مستقل از سایر واحدها می‌باشد و دارای تغذیه، وظیفه و عملکرد مربوط به خود است. هر واحد از طریق درگاه‌های ورودی خروجی با سایر واحدها در ارتباط است. طراحی ماژولار باعث می‌شود عیب‌یابی، تغییر و ارتقاء مدار به آسانی انجام شود.[۹]




*شکل ۲- ۲۴ طراحی ماژولار، در این طراحی مدار به بخش‌هایی مستقل به نام واحد تقسیم بندی می‌شود. عملکرد هر واحد مستقل از سایسر واحدها است. برای مثال در این شکل مدار دارای ۵ واحد است. مرز دقیق و شفافی میان واحدها وجود دارد.*

### ۲-۶-۱۲ بررسی شیوه‌های پیاده‌سازی سخت افزاری الگوریتم

در این بخش دو فرضیه مطرح شد: پردازش اطلاعات توسط مدار روبات در روی روبات و پردازش اطلاعات توسط یک عامل خارجی در بیرون روبات. در حالتی که پردازش اطلاعات توسط یک عامل خارجی انجام می‌شود، ارتباط روبات با عامل خارجی به صورت ارسال اطلاعات ورودی و دریافت فرمان می‌باشد. در ادامه به بررسی مزایا و معایب هر یک از این دو روش در مقایسه با یکدیگر می‌پردازیم.

- پردازش الگوریتم اصلی روی مدار:
  - مزایا:
    - کاهش هزینه‌ی پردازش
    - نزدیک بودن بیشتر به ماهیت "نهفتگی سیستم" (Embedded System)
    - کاهش توان مصرفی روبات
    - کاهش تاخیر ارسال و دریافت اطلاعات
    - افزایش امنیت اطلاعات
    - جلوگیری از نفوذ مزاحم به سیستم کنترلی روبات
    - عدم نیاز به پیاده‌سازی ابزارهای امنیت شبکه
    - عدم نیاز به رمزنگاری و رمزگشایی از اطلاعات
    - کاهش تاثیر نویز بر رسانه‌های ارتباطی روبات
    - عدم تداخل کانال‌های روبات‌ها با یکدیگر
  - معایب:
    - سرعت پایین‌تر پردازش به دلیل محدودیت فرکانسی پردازنده‌ها
    - محدودیت برنامه‌نویسی با زبان های سطح پایین و متوسط
    - دشواری تغییر و عیب یابی برنامه
    - عدم امکان پیاده‌سازی یک برنامه با چندین زبان بر روی یک پردازنده (به آسانی)
    - محدودیت حافظه

- پردازش الگوریتم توسط عامل خارجی:
  - مزایا:
    - سرعت بالای پردازش به دلیل عدم محدودیت فرکانسی
    - عدم محدودیت در سطح زبان برنامه‌نویسی
    - امکان پیاده‌سازی الگوریتم بوسیله ی چندین زبان
    - امکان پیاده‌سازی شی‌گرا
    - استانداردسازی برنامه
    - سهولت ارتقاء، توسعه و عیب‌یابی برنامه
    - امکان استفاده آسان از عیب‌یابی قدم به قدم (Step by step)
    - عدم محدودیت حافظه
  - معایب:
    - کاهش هزینه‌ی پردازش
    - دور شدن از ماهیت "نهفتگی سیستم" (Embedded System)
    - افزایش توان مصرفی روبات
    - افزایش تاخیر ارسال و دریافت اطلاعات
    - کاهش امنیت اطلاعات
    - امکان نفوذ مزاحم به سیستم کنترلی روبات
    - نیاز به پیاده‌سازی ابزارهای امنیت شبکه
    - نیاز به رمزنگاری و رمزگشایی از اطلاعات
    - افزایش تاثیر نویز بر رسانه‌های ارتباطی روبات
    - امکان تداخل کانال‌های روبات‌ها با یکدیگر

### ۲-۶-۱۳ انتخاب شیوه‌ی مناسب پردازش سخت‌افزاری الگوریتم

به دلیل ماهیت سکو بودن پروژه، پردازش الگوریتم توسط عامل خارجی مناسب‌تر است و برتری محسوسی نسبت به پردازش الگوریتم روی مدار دارد. زیرا ماهیت سکو بودن پروژه ایجاب می‌کند تا محدودیتی از جهت فرکانس و سرعت پردازش، سطح و نوع زبان مورد برنامه‌نویسی و میزان حافظه‌ی مورد استفاده برای الگوریتم وجود نداشته باشد. همچنین به سبب ماهیت سکو بودن پروژه، برنامه‌نویس باید قادر باشد تا با سهولت به ارتقاء، تغییر و عیب‌یابی قدم به قدم برنامه بپردازد. با مقایسه‌ی این شرایط با آنچه در بخش ۲-۶-۱۲ عنوان شد، مشخص می‌شود که باید از شیوه‌ی پردازش اطلاعات توسط عامل خارجی استفاده کرد.

### ۲-۶-۱۴ بررسی حسگرهای پیشنهادی

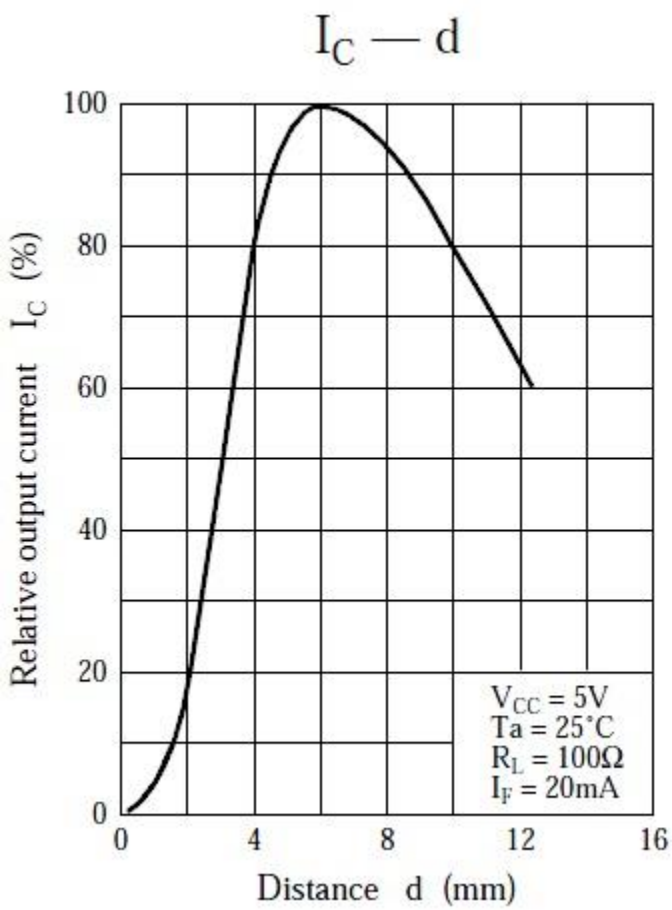
در این بخش همانند سایر بخش‌ها، به بررسی ویژگی های هر یک از حسگرهای روبات که در بخش ۲-۵ مطرح شدند، پرداخته می‌شود. دو خانواده‌ی حسگرهای فراصوت و مادون‌قرمز با یکدیگر به شرح زیر مقایسه می‌گردند:

- خانواده‌ی حسگرهای مادون قرمز:

- مزایا:
  - انتشار امواج به خط راست و درنتیجه خطای تشخیص مانع پایین
  - قیمت نسبتا پایین (به طور متوسط ۱ دلار آمریکا)
  - تنوع بیشتر در بازار
  - مصرف توان پایین حدود ۳۰ میلی‌آمپر به طور متوسط
  - توانایی تشخیص رنگ موانع
  - راه اندازی بسیار ساده: آشکارسازی مانع به صورت یک اختلاف پتانسیل متغیر در خروجی حسگر

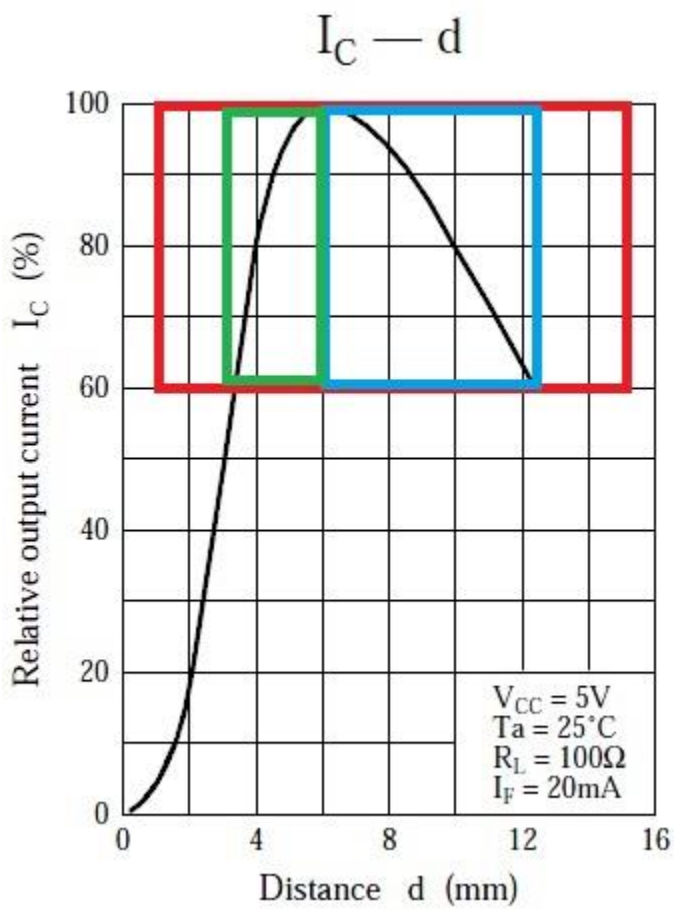
- معایب:
  - برد عملیاتی کم (حداکثر ۲ سانتی‌متر) [۲۰-۲۲]
  - حساسیت به نور محیط
  - نیازمند کالیبره کردن حسگر مطابق با نور محیط
  - حساسیت به سایر تجهیزات مادون قرمز مانند دوربین‌ها، نور فلش و غیره
  - یک به یک نبودن نمودار ولتاژ/جریان-فاصله
  - عمومیت نداشتن بسته‌بندی به صورت ماژول در این خانواده

با وجود معایب ذکر شده در بالا، امروزه برای بسیاری از کاربردها از حسگرهای مادون قرمز استفاده می‌شود. با استفاده از بعضی روش‌ها و برای بعضی از کاربردها می توان برد این حسگرها را افزایش داد. برای مثال شکل ۲-۲۵ نمودار تشخیص فاصله‌ی حسگر مادون قرمز ON2179 را نشان می دهد. همانطور که ملاحظه می‌شود، برد این حسگر برای تشخیص یک مانع حداکثر ۱۲ میلی‌متر می باشد. اما با کم‌کردن تاثیر نور محیط با استفاده از آرایش تفاضلی این حسگر و همچنین کاهش اثر ماده بازتاب کننده، برد نزدیک به ۴۰ سانتی‌متر برای آن گزارش شده است.[۲۳]



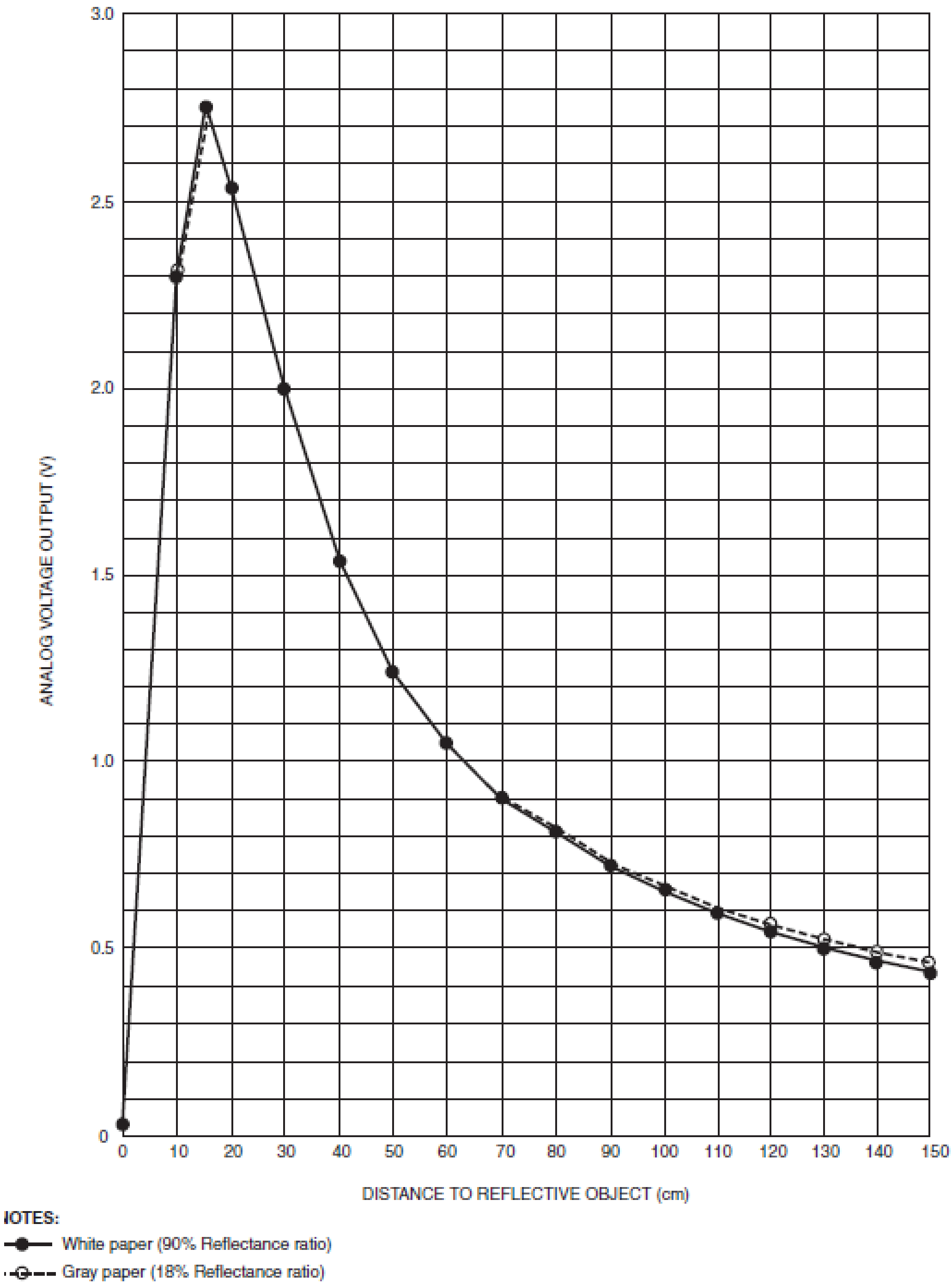

*شکل ۲- ۲۵ نمودار تشخیص فاصله حسگر ON2179، در حالت کلی بهترین برد این حسگر مادون قرمز مطابق با نمودار ۱۲ میلی‌متر می‌باشد.*

عیب دیگری که در نمودار اغلب حسگرهای مادون قرمز مشاهده می شود، یک به یک نبودن نمودار می‌باشد.[۲۴] شکل ۲-۲۶ همان نمودار تشـخیص فاصـله حسـگر ON2179 را دوباره نشـان می‌دهد. همانطور که در این شـکل دیده می‌شـود، بخشی از نمودار که با رنگ قرمز مشـخص شـده است یک به یک نمی‌باشد. مفهوم این موضوع این است که برای فاصله های میان ۴ تا ۱۲ سانتی‌متر، حسگر هیچ گونه تمایزی میان بخش سبز و آبی نمودار قائل نمی‌شود. برای مثال اختلاف پتانسیل خروجی حسگر برای ۴ و ۱۲ سانتی‌متر یکسان است.

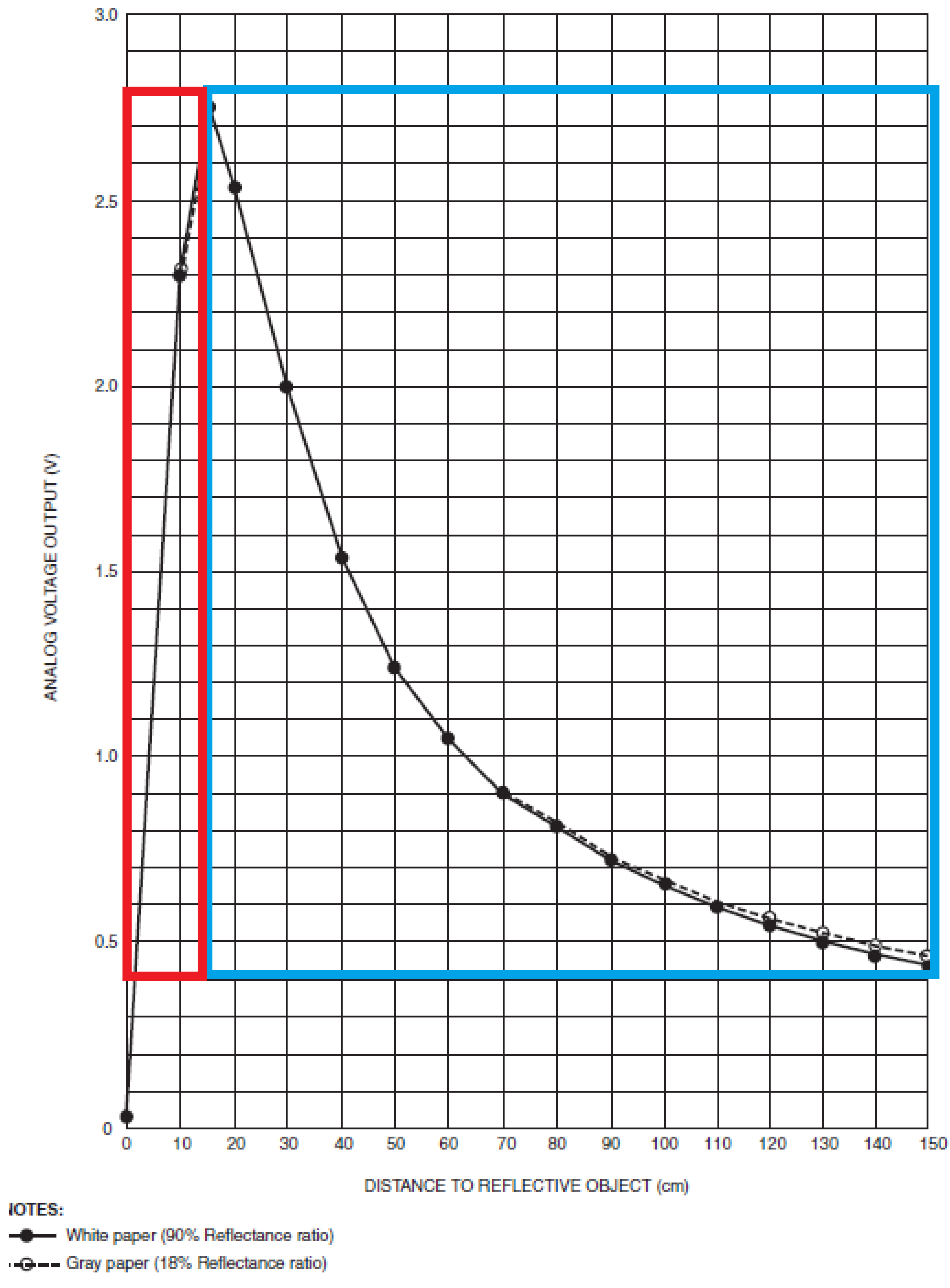



*شکل ۲- ۲۶ ناحیه ی قرمز رنگ نمودار حسگر مادون قرمز ON2179 یک به یک نمی‌باشد، حسگر مادون قرمز در این ناحیه دچار خطا می شود*

امروزه بهینه سازی‌های فراوانی بر روی حسگرهای مادون‌قرمز انجام شده است. عمده این بهینه سازی‌ها در زمینه‌ی تبدیل حسگرهای مادون قرمز به ماژول‌هایی اسـت که به صـورت یک بسـته‌بندی مسـتقل قادر به اندازه‌گیری فاصله یا تشخیص مانع باشند. شکل ۲-۲۷ نمودار مشخصه ماژول مادون قرمز GP2Y0A02YK ساخت شرکت شارپ می‌باشد.[۲۵] همانطور که در این شکل ملاحظه می شود، برد این حسگر تا ۱۵۰ سانتی‌متر افزایش یافته است. این رقم افزایش، بردی معادل ۱۰۰ برابر میانگین برد حسگرهای مادون‌قرمز معمولی می باشد. با وجود این سـطح از بهینه‌سـازی، اما هنوز مشـکل یک به یک نبودن نمودار مشـخصـه این حسگرها برطرف نشده است. زیرا همانطور که در شکل ۲-۲۸ ملاحظه می شـود، خروجی ماژول برای نقاط متناظر محدوده‌ی سـبز و آبی که دارای عرض یکسـانی هستند برابر است. یعنی به عنوان مثال خروجی ماژول برای فواصل ۱۳۰ سانتی‌متر و ۳ سانتی‌متر یکسان است. این عدم تمایز میان نقاط، باعث می‌شود دقت ماژول بسیار کاهش یابد.




*شکل ۲-۲۷ نمودار مشخصه حسگر GP2Y0A02YK، این ماژول ساخت کارخانه‌ی شارپ می‌باشد. همانطور که در نمودار ملاحظه می‌شود، در برد این حسگر مادون‌قرمز ۱۵۰ سانتی‌متر است که بسیار بیشتر از حسگرهای مادون قرمز معمولی مانند ON2179 می‌باشد.*




*شکل ۲- ۲۸ در حسگر حسگر GP2Y0A02YK به ازای هر نقطه در ناحیه آبی، یک نقطه هم عرض در ناحیه قرمز وجود دارد، بنابراین حسگر میان این دو نقطه تمایزی قائل نمی شود. در واقع هر نقطه در محدوده‌ی آبی رنگ یک نقطه‌ی مزدوج در ناحیه‌ی قرمز رنگ دارد. خروجی حسگر به ازای هر دو نقطه‌ی مزدوج یکسان است.*

- خانواده ی حسگرهای فراصوت:

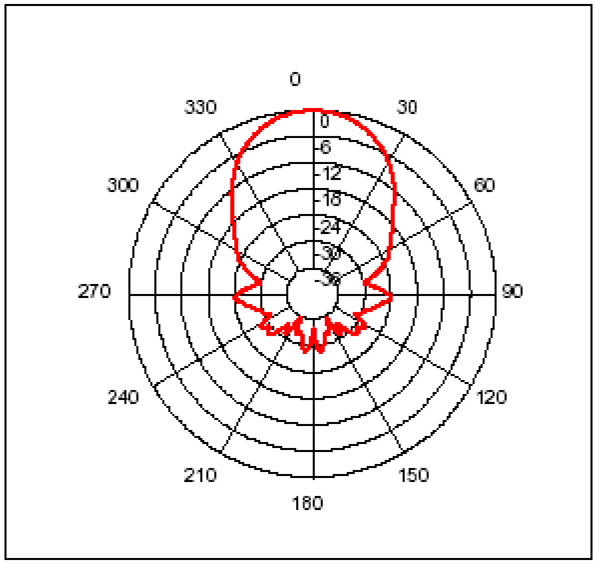
  - مزایا:

    - برد عملیاتی بسیار بالا(بین ۳ تا ۴ متر)
    - موجود بودن حسگر به صورت ماژول
    - عدم حساسیت به نویز یا عوامل محیطی
    - عدم نیاز به کالیبریزاسیون مجدد قبل از هر بار استفاده
    - عدم وجود مشکل یک به یک نبودن نمودار مشخصه که در حسگرهای مادون قرمز بحث شد

  - معایب:

    - انتشار امواج به صورت مخروطی و در نتیجه کاهش دقت تشخیص موانع
    - قیمت به نسبت بالا (به طور متوسط ۸۰ دلار آمریکا)
    - راه انـدازی نســـبتا پیچیده به دلیل پشـــتیبانی از پروتکل هایی مانند I2C که در راه‌اندازی مشکلات فراوانی ایجاد می‌کنند
    - مصرف توان نسبتا زیاد (به طور متوسط حداقل ۱۰۰ میلی‌آمپر عملیاتی به ازای هر حسگر)
    - امکان تداخل امواج ماژول‌های مجاور با یکدیگر و در نتیجه ایجاد خطا
    - امکان تداخل امواج ماژول‌های روبات‌ها متفاوت با یکدیگر و در نتیجه ایجاد خطا

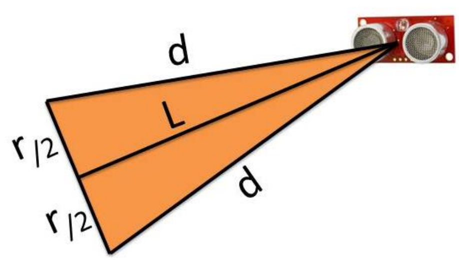
همانطور که در بالا اشاره شد، یکی از مهمترین معایب ماژول‌های فراصوت الگوی انتشار امواج آنها می‌باشد، مطابق شکل شماره ۲-۲۹، الگوی انتشـار امواج در این خانواده از حســگرها، یک الگوی مخروطی اســت. با توجه به اینکه سطح مقطع مخروط با افزایش برد و دور شدن از ماژول به طور خطی افزایش می‌یابد، خطای تشخیص در این ماژول‌ها زیاد می‌شود. برای مثال در شکل ۲-۲۹، هر جسمی که ماژول در فاصله‌ی d تشــخیص دهد، در محدوده ای با زاویه ی ۳۰+ درجه تا ۳۰ – از مرکز ماژول قرار دارد. به عبارتی دیگر حســـگر در مورد محل دقیق مانع یا جسم دچار عدم قطعیت می شود.




*شکل ۲- ۲۹ الگوی امواج حسگر فراصوت در دو بعد، این الگوی انتشار امواج در سه‌بعد طرحی شبیه به مخروط ایجاد می‌نماید.*

$\sin\alpha = r_{/2} \div d \qquad r = 2 * \sin\alpha_{/d}$

$\cos\alpha = L_{/d} \qquad L = d * \cos\alpha$




*شکل ۲- ۳۰ محدوده مقادیر معتبر حسگر فراصوت، با توجه به اینکه الگوی انتشار امواج فراصوت به صورت یک مخروط سه‌بعدی است، محدوده‌ای ار مقادیر معتبر به جای صرفا یک مقدار، برای یک مانع مشخص گزارش می‌شود.*

شـکل ۲-۳۰، محدوده مقادیر معتبر حسـگر فراصـوت و همچنین میزان خطای آنرا در تشخیص یک مانع نشان می‌دهد. مطابق با این شکل، چنانچه حسگر فراصوت یک مانع در فاصله‌ی d سانتی متری از خود را تشخیص دهد، این مانع در بازه‌ای خطی به مرکز L و با شروع از r/2– و خاتمه به r/2+ قرار دارد. با این حساب، محدوده‌ی مقادیر معتبر برای مانع (یعنی بازه‌ای که مانع در آن قرار دارد) از معادله های ۲-۴ و ۲-۵ بدست می‌آید.

معادله ۲-۴ $-r_{/2} < X_{Rang} < +r_{/2}$

معادله ۲-۵ $L < Y_{Rang} < d$

در معادله ۲- ۴ ، $X_{Range}$ محدوده‌ی مقادیر معتبر برای مختصات عرض مانع می باشد. همچنین r محدوده‌ی شعاع حسگر می‌باشد و از شکل ۲-۳۰ محاسبه می‌شود. در معادله ۲-۵، $Y_{Range}$ محدوده مقادیر معتبر برای مختصات عرض مانع است. مطابق با این معادلات، هرچه مانع در فاصله‌ی بیشتری از حسگر قرار داشته باشد، عدم قطعیت حسگر در تشخیص و گزارش مختصات دقیق آن، به دلیل وسعی تر بودن بازه‌های موجود در معادلات ۲-۴ و ۲-۵، بیشتر است.

## ۲-۶-۱۵ انتخاب حسگر مناسب

با توجه به اینکه این روبات سکو با هدف برطرف‌کردن مشکل جمع آوری اطلاعات روبات‌های هوشمند دارای حسگر و ارتقاء نظام جمع آوری اطلاعات در آنها، طراحی شده است، بنابراین انتخاب حسگرها به عنوان واسط میان روبات و محیط خارج از آن، از اهمیت بالایی برخوردار می‌باشد. در انتخاب حسگر مناسب، با یک رقابت جدی میان حسگرهای فراصوت و مادون قرمز روبرو هستیم.

حسگرهای مادون قرمز به دلیل انتشار امواج مادون قرمز به خط راست دارای دقت بیشتری می‌باشند، اما برد عملیاتی آنها بسیار پایین است (در حدود ۱ تا ۲ سانتی‌متر). همانطور که در بخش ۲-۶-۱۶ نیز بحث شد، این حسگرها در تشخیص مانع و گزارش فاصله تا آن دارای عدم قطعیت می‌باشند و به جای یک فاصله ی معین و یکتا دو فاصله با اختلاف بسیار زیاد را گزارش می‌کنند. برای مثال یک مانع ثابت و یکتا را در فواصل ۳ و ۳۰ سانتی‌متری از خود تشخیص می‌دهند، اما مشخص نمی کنند دقیقا در کدام یک از دو نقطه قرار دارد.در مقابل حسگرهای فراصوت نویزپذیر نیستند و برد بسیار بالایی دارند (حداقل ۱ متر به صورت عملیاتی، حداکثر بین ۳ تا ۴ متر). اما به شکل دیگری عدم قطعیت دارند. عدم قطعیت آنها به صورت گزارش مانع در یک بازه مختصات است. در این حالت مانع می‌تواند در هریک از نقاط آن بازه وجود داشته باشد.

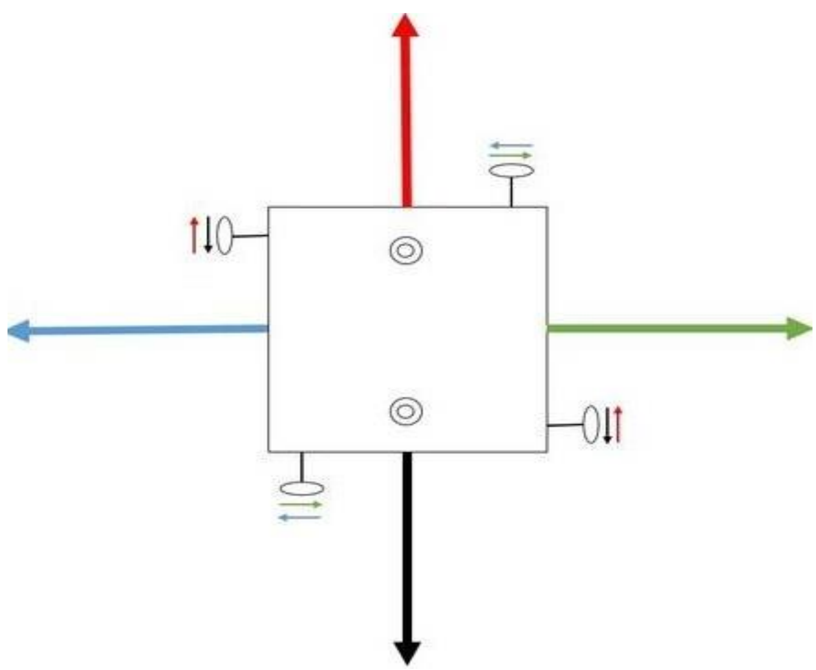
با توجه به اینکه عدم قطعیت حسگرهای فراصوت به صورت بازه‌ای می باشد و این بازه مطابق معادلات ۲-۶ و ۲-۷ با محدود کردن برد معتبر حسگر قابل کنترل است و همچنین به خاطر عدم نویزپذیری و برد عملیاتی بالا، خانواده ی حسگرهای فراصوت به عنوان انتخابی مناسب برای حسگر روبات سکو انتخاب می‌شود. البته علاوه بر این مزایا، همانطور که در ابتدای بخش ۲-۶-۱۴ گفته شد، این خانواده از حسگرها، یکی از محبوب ترین و پرطرفدارترین خانواده حسگرها در کشور می‌باشند و به آسانی و با تنوع فراوان در بازار داخلی یافت می‌شوند. به این ترتیب انتخاب خانواده‌ی حسگرهای فراصوت به عنوان حسگر روبات، شرط بومی بودن روبات را نیز ارضاء می‌نماید.

## ۲-۷ جمع‌بندی فصل دوم

در این فصــل با هدف طراحی یک روبات ســکو که بتواند مشــکل جمع‌آوری اطلاعات را برای اجرای الگوریتم‌هایی مانند جســتجو، پویش و نقشــه‌برداری برطرف نماید، ایده‌پردازی‌های مختلف انجام شــدند. این ایده‌ها در ابتدای فصــل در زمینه‌های مکانیک وشــاســی، الکترونیک و شــیوه‌های پردازش سـخت‌افزاری و نرم‌افزاری الگوریتم مطرح شدند. هر یک از ایده ها تا حد امکان در ابعاد تجربی و آزمایشگاهی پیاده سازی شـدند و معایب و مزایای آنها مورد بررسی قرار گرفتند. در هر بخش، پس از بررسی هر ایده، به جمع بندی نکات آن پرداخته شد و ایده‌ی برتر در هر زمینه شـناسـایی و معرفی شـد. هر یک از این ایده‌ها فقط در بعد کلی مطرح شدند و از ورود به جزئیات تا حد امکان خودداری شد. در نهایت، ویژگی‌های روبات سکوی مورد نظر با در نظر گرفتن تمامی شرایط مطرح شده در فصل ۱ به صورت زیر خلاصه شد:

- آرایش مکانیکی:

مطابق شـکل ۲-۲۵، آرایش مکانیکی شامل ۴ موتور که دو موتور در راستای x و دو موتور در راستای y قرار دارد، می‌باشــد. همچنین دو عدد حســگر بر روی دو عدد موتور در بالا و پایین روبات قرار دارند که با حرکت خود ۳۶۰ درجه اطراف روبات را پوشش می‌دهد.



*شکل ۲- ۳۱ آرایش مکانیکی انتخاب شده برای روبات سکو، چهار دایره‌ی کوچک در چهار سمت روبات معرف چرخهای حرکتی روبات هستند. این چرخها همگی خورشیدی می‌باشند. زاویه‌ای که چرخهای راستای افق با چرخهای راستای قائم می‌سازند ۹۰ درجه است. دایره هایی که در داخل روبات در سمت بالا و پایین قرار دارند، حسگرها و موتورهای گرداننده‌ی حسگر را نشان می‌دهند.*

- جنس شاسی: پلکسی گلاس
- موتورهای حرکتی روبات: DC
- موتورهای گرداننده ی حسگر: پله‌ای
- نوع حسگر: فراصوت
- شیوه ی پردازشی اطلاعات: پردازش متمرکز
- ساختار مدار اصلی روبات: ماژولار
- شیوه پیاده سازی سخت‌افزاری الگوریتم: پردازش توسط یک عامل خارجی

در فصل سوم به پیاده سازی Platform موردنظر مطابق با شرایط بالا و با ذکر جزئیات ساخت خواهیم پرداخت.

# فصل سوم

# جزئیات و پیاده‌سازی‌ها

## ۳-۱ طرح کلی فصل

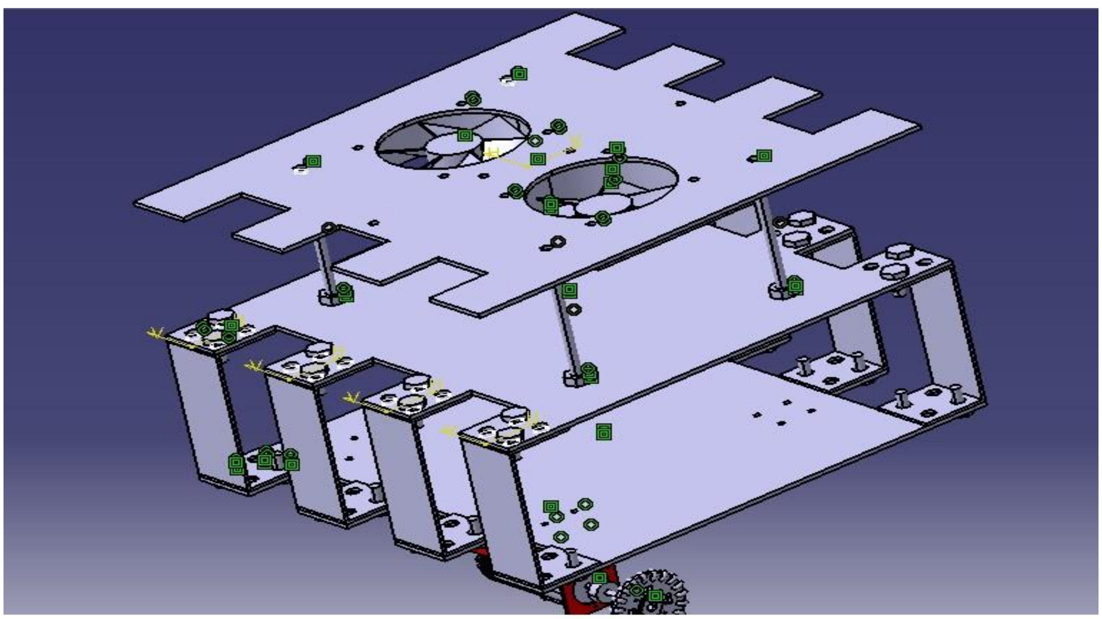
در این فصل به طور عمده به جزئیات روبات و همچنین پیاده‌سازی بخش‌ها و اجزای مختلف آن پرداخته می‌شود. در بحث جزئیات، در مورد هر بخش، تا حد امکان جهت شفاف شدن موضوع بحث خواهد شد و پیاده سازی‌ها جزء به جزء بررسی خواهند شد. هدف از این فصل به طور خاص، آشـنایی بیشتر با جزئیات ساخت روبات سکویی است که در فصل اول به ارائه ی معیارهای طراحی آن پرداخته شد و در فصل دوم نمایی کلی از آن مشـخص شـد. همچنین با توجه به ویژگی های ظاهری روبات که در فصـل ۲ مشـخص شـدند، از این پس روبات را DSPR می‌نامیم. این نام به طور مخفف از ابتدای کلمات Dual Sensor Platform Robot گرفته شـده اسـت و به معنای روبات سـکوی دو حسـگره می‌باشد.

## ۳-۲ پیاده‌سازی‌های مکانیکی

در این بخش، جزئیات پیاده سـازی های مکانیکی DSPR، مطابق با ایده‌هایی که در فصـل ۲ انتخاب شـدند و به عنوان ایده های برتر شناخته شدند، مطرح می‌گردد.

### ۳-۲-۱ پیاده‌سازی شاسی DSPR

شـکل ۳-۱ چشـم اندازی ۳ بعدی از شـاسـی روبات را که با نرم افزار Catia طراحی شـده اسـت نشان می‌دهد. جنس این شاسی از پلکسی گلاس با ضخامت ۳ میلی‌متر می باشد. همانطور که در این شکل مشاهده می‌شود، روبات مجموعا از سه صفحه و دو طبقه تشکیل شده اسـت. این صـفحه‌ها به ترتیب از پایین به بالا صفحات یک، دو و سه نام گذاری می‌شوند. صفحه شماره یک، اتصالات چرخ، موتورها و باتری را حمل می‌کند. در صفحه ی شماره ۲، مدار اصلی روبات قرار دارد. صفحه ی شماره ۳ سقف روبات است و با هدف حفاظت از مدار روبات طراحی شده است. این صفحه همچنین دارای دو عدد فن خنک کننده جهت خنک کردن قطعات گرماده مدار می‌باشد.



شکل ۳- ۱ پرسپکتیو DSPR، روبات مجموعا از سه عدد صفحه‌ی پلکسی‌گلاس با ضخامت ۳ سانتی‌متر ساخته‌شده است.

### ۳-۲-۲ پیاده‌سازی انکودرهای DSPR

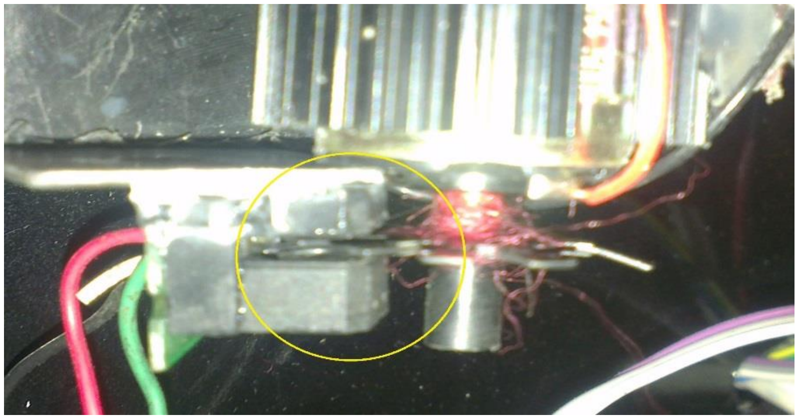
روبات برای تشـخیص مختصـات خود و مسـافت پیموده شده، باید دارای انکودر(Encoder) بر روی موتورهای خود باشد. انکودر میزان مسافت طی شده توسط هر موتور را نشان می‌دهد. چون دو موتور روبات در راستای محور xها و دو موتور دیگر در راستای محور yها می‌باشند بنابراین با قرار دادن یک انکودر بر روی موتور محور x ها و یک انکودر بر روی محور yها می‌توان در هر لحظه مختصات روبات را بدست آورد.

برای موتورهای حرکتی روبات، پس از انجام جســتجو در میان محصــولات موجود در بازار بومی، موتورهای L012CR2342 ســاخت شرکت FAULHABER انتخاب شدند. این موتورها دارای سرعت ۱۲۰ دور در دقیقه و گشتاور عملیاتی ۹,۲ کیلوگرم بر سانتی‌متر (اندازه گیری شـده در آزمایشـگاه به صـورت تجربی توسط مولفین پایان‌نامه) می‌باشند. آنچه به این موتورها نسبت به سایر موتورها برتری می‌بخشد گشتاور عملیاتی، مصـرف جریان پایین تقریبا ۱۰۰ میلی آمپر به طور عملیاتی، مجهز بودن به پره‌های انکودر و قیمت به نسبت پایین آنها در مقایسه با موتورهای مشابه می‌باشد. شکل ۳-۲ نمایی از موتورهای حرکتی روبات را نشان می‌دهد. این طرح با استفاده از نرم افزار Catia و با ابعاد واقعی موتور طراحی شـده اسـت. همانطور که ملاحظه می‌شـود در پشت موتور پره هایی قرار دارد که متناسب با تعداد مشخصی از گردش دور موتور می‌چرخند.



*شکل ۳- ۲ موتور حرکتی روبات و پره های انکودر آن*

با قرار دادن یک شـمارنده‌ی نوری در میان پره‌های انکودر موتور و شـمارش تعداد پره‌هایی که در طی حرکت روبات از میان شمارنده عبور می‌کنند، می توان مسـافت پیموده شده توسط موتور را محاسبه کرد. شکل ۳-۳ نمایی نزدیک از پیاده‌سازی شمارنده‌ی نوری را در میان پره‌های انکودر نشان می‌دهد.



شکل ۳- ۳ پیاده سازی انکودر روبات، دایره‌ی زرد رنگ یک عدد شمارنده‌ی نوری را نشان میدهد که در میان پره‌های انکودر متصل به موتور قرار گرفته است. با چرخش موتور و در نتیجه چرخش این پره‌ها، شمارنده‌ی نوری قادر به شمارش تعداد دورهایی است که موتور چرخیده است.

### ۳-۲-۳ پیاده‌سازی موتورهای گرداننده ی حسگر

همانطور که در فصل دو مشخص شد، برای گردش حسگرهای روبات از موتور سرو استفاده می‌شود. همچنین حسگرهای روبات حسگرهای فراصوت می‌باشند که بر روی موتور سرو قرار می‌گیرند. شکل ۳-۴ نحوه‌ی قرارگیری حسگرهای روبات بر روی موتور سرو را نشان می‌دهد. همانطور که در این تصویر مشاهده می شود، هر یک از حسگرهای فراصوت در داخل یک محفظه‌ی تفلونی قرار گرفتند و بر روی سرو موتور نصب شده اند.



شکل ۳- ۴ پیاده سازی قرارگرفتن حسگرهای فراصوت بر روی سرو، هریک از حسگرهای فراصوت بر روی یک محفظهی تفلونی قرارگرفته‌اند و بر روی روبات نصب شده‌اند.

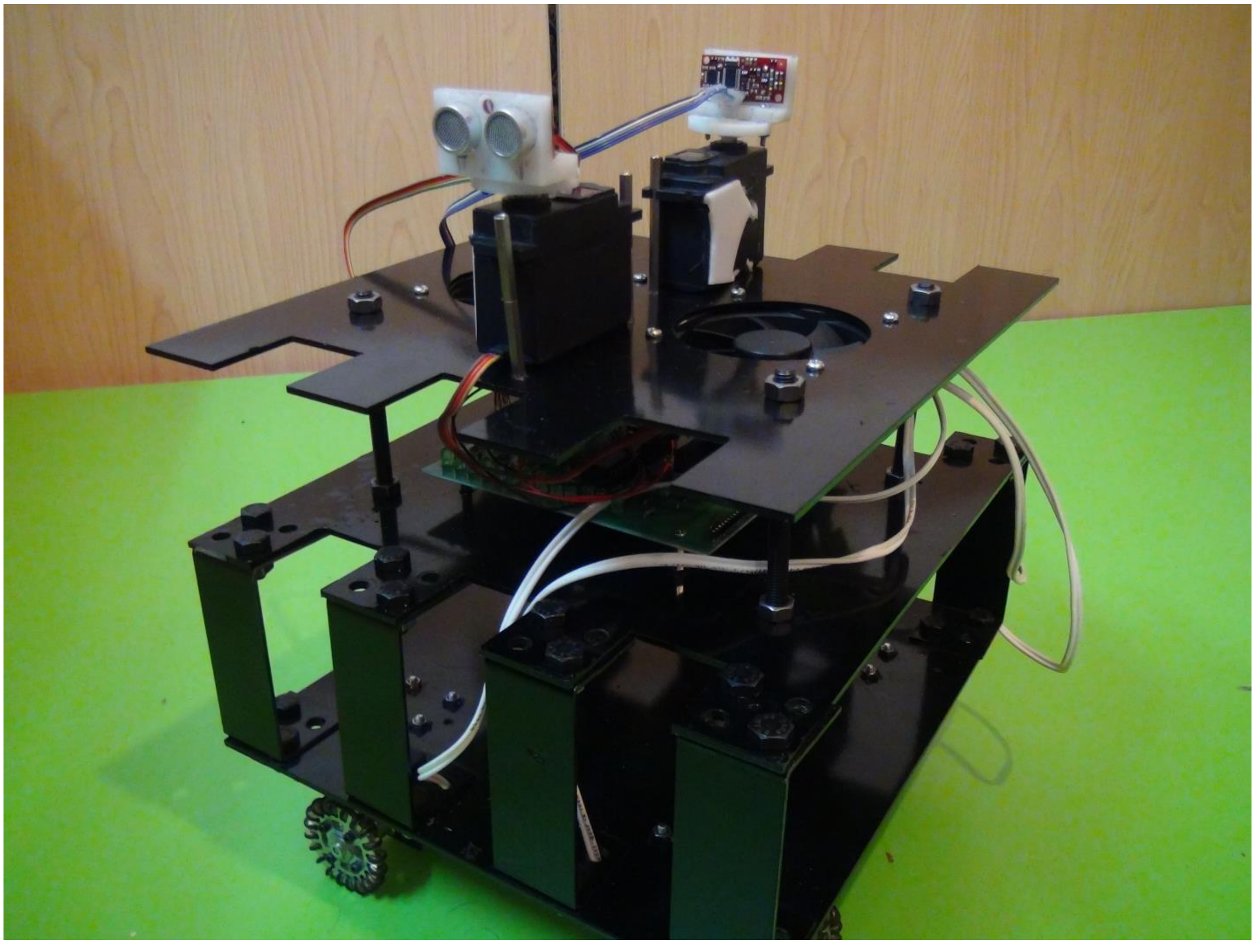
در میان سـرو موتورهای موجود در بازار بومی، سـرو موتور 9505MG انتخاب شـد. شـکل ۳-۵ مشخصات فنی این سرو موتور را نشان می‌دهد. دلیل انتخاب این سـرو موتور گشتاور بالای آن (بین ۲۰ تا ۲۵ کیلوگرم)، امکان کنترل گشتاور به وسیله ی اختلاف پتانسیل و از همه مهمتر سرعت عملیاتی بسیار بالای آن می‌باشد. این سرو موتور از محبوب‌ترین و پرفروش‌ترین سرو موتورها در بازار بومی کشور است.

| 9805MG | metal gear |
| --- | --- |
| ابعاد | 66*30.2*64.4 میلی متر |
| گشتاور را انداز | 25kg/cm(6V   (20kg/cm(4.8V ) |
| سرعت عملیاتی | 0.20sec/60degree(4.8v), 0.16sec/60degree(6v) |
| ولتاژ کاری | 4.8-7.2V |
| دمای کاری | 0℃_ 55℃ |
| پهنای باند مرده | 5us |
| وزن | 160 گرم |

*شکل ۳- ۵ مشخصات فنی سرو موتور 9505 MG، همانطور که ملاحظه می‌شود، این سرو موتور دارای سرعت عملیاتی پیمایش ۶۰ درجه درحدود ۲۰۰ میلی‌ثانیه و گشتاور حدود ۲۰ کیلوگرم بر سانتی‌متر می‌باشد.*

### ۴-۲-۲ تصویر پیاده‌سازی کامل مکانیک و شاسی DSPR

شکل ۳-۶ تصویری از بدنه‌ی کامل روبات DSPR را نشان می‌دهد. همانطور که در این شکل ملاحظه می‌شود، دو عدد حسگر فراصوت بر روی دو عدد موتور سـرو قرار داده شـده‌اند. همچنین ۴ عدد موتور DC در زیر روبات به ۴ عدد چرخ خورشیدی متصل شده است و سیستم حرکتی روبات را تشـکیل داده اند. همانطور که پیش تر توضـیح داده شد، موتورهای DC با آرایش خاصی در زیر روبات قرار دارند. دو موتور در راسـتای محور xها و دو موتور در راسـتای محور yها می‌باشند. بدین ترتیب حرکت روبات در هر لحظه فقط در راستای یکی از دو محور x یا y می‌باشد. به عبارت دیگر در هر لحظه از حرکت یا فقط x روبات تغییر می‌کند و یا فقط y آن و نه هردو آنها به صورت همزمان.



*شکل ۳- ۶ نمونه‌ی مکانیکی کامل ساخته شده‌ی DSPR با جزئیات تمام*

## ۳-۳ انتخاب پردازنده‌های DSPR

در انتخاب سیستم پردازش، میان پردازنده های مختلف با معماری‌های متفاوت، میکروکنتلرهای AVR انتخاب شدند. دلیل اصلی انتخاب این میکروکنتلرها، قیمت بسیار مناسب آنها و یافت شدن آسان در بازار بومی داخلی بود. همچنین منابع لازم برای راه‌اندازی آنها به فور به زبان پارسی موجود می‌باشد. این خانواده از میکروکنتلرها در میان جامعه‌ی روباتیک ایران بسیار محبوب می‌باشند. میکروکنتلرهای AVR تقریبا از تمامی پروتکل‌های مورد نیاز برای DSPR پشتیبانی می‌کنند. برای مثال دارای تولید کننده‌ی پالس PWMجهت راه‌اندازی موتورهای سرو می‌باشند، از پروتکل I2C و UART جهت راه‌اندازی حسگرها و ارتباط با یکدیگر پشتیبانی می‌کنند. این میکروکنتلرها دارای تولید کننده‌ی پالس داخلی هستند و برای بسیاری از کاربردهای خود نیاز به منبع پالس خارجی ندارند و قادر هستند فرکانسی میان ۱ تا ۱۶ مگاهرتز تولید کنند.[۱۱] همچنین برخلاف میکروپروسسورها یا FPGA ها، نیازمند مدارات جانبی مانند تولید کننده پالس، بافر خروجی، بافر ورودی و حافظه نمی‌باشند.

در بخش ۲-۶-۱۳ در مورد شیوه پردازش اطلاعات روبات بحث شد و در نهایت شیوه پردازش توزیع شده انتخاب شد. در این شیوه پردازشی، هر بخش توسط یک ریزپردازنده پردازش می‌شود. همه ی بخش‌ها با یکدیگر در ارتباط هستند. در DSPR بخش‌های مجزا عبارتند از: حسگرفراصوت، موتور سرو، انکودر و پردازنده‌ی اصلی. بنابراین حداقل به ۵ عدد میکروکنتلر(دو عدد برای حسگرها) نیاز است.[۲۶]

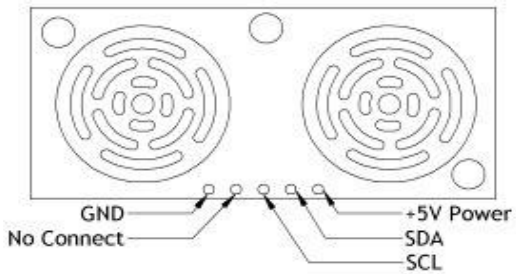
## ۳-۴ انتخاب و راه اندازی قطعات الکترونیک پروژه و طراحی مدار آنها

### ۳-۴-۱ انتخاب حسگر فراصوت

با انجام جستجو در میان حسگرهای فراصوت موجود در بازار بومی و مطالعه‌ی مشخصات فنی آنها، حسگر فراصوت SRF08 انتخاب شد. شکل ۳-۷ نمایی از این حسگر فراصوت را نشان می‌دهد. همانطور که در این شکل که ملاحظه می شود، این حسگر فراصوت به صورت ماژول موجود می باشد. حداقل برد مناسب ۳ سانتی‌متر، حداکثر برد ۶ متر، امکان تنظیم دقت، اندازه‌گیری فاصله به دفعات قبل از گزارش فاصله، پشتیبانی از پروتکل I2C و مصرف جریان بسیار پایین ۱۵ میلی‌آمپر دلایل اصلی انتخاب این حسگر در میان تنوع حسگرهای موجود در بازار داخلی کشور می‌باشند. شکل ۳-۸ سایر مشخصات فنی این حسگر را نمایش می دهد. شکل ۳-۹ نقشه ی پایه های ماژول حسگر SRF08 را مشخص می کند.[۲۷]



*شکل ۳- ۷ ماژول حسگر فراصوت SRF08*




*شکل ۳- ۸ نقشه ی پایه های ماژول SRF08 این ماژول جمعا دارای ۵ پایه می‌باشد: دو پایه برای تغذیه، دو پایه برای پشتیبانی از پروتکل I2C و یک پایه برای پیکربندی*

| Specifications | |
|---|---|
| Voltage | 5 V |
| Current | 15 mA Typ. 3 mA Standby |
| Frequency | 40 kHz |
| Minimum Range | 3 cm |
| Maximum Range | 6 m |
| Max Analogue Gain | Variable to 1025 in 32 steps |
| Connection Type | I2C |
| Light Sensor | Front facing light sensor |
| Timing | Fully timed echo, freeing host computer of task |
| Echo | Multiple echo - keeps looking after first echo |
| Units | Range reported in µS, mm or inches |
| Weight | 11.3 g (0.40 oz) |
| Size | 43 x 20 x 17 mm (1.69 x 0.78 x 0.67 in) |

*شکل ۳- ۹ جدول مشخصات فنی ماژول SRF08*

### ۳-۴-۲ راه اندازی حسگر فراصوت

برای راه اندازی حسگر فراصوت SRF08 از میکروکنتلر ATMega 8 استفاده شد. این میکروکنتلر از خانواده ی AVR و از زیرشاخه ی Mega می‌باشد و دارای فضای آدرس دهی ۸ بیتی و حافظه ی Flash ۸ کیلوبایتی است. دلیل انتخاب این میکروکنتلر توان مصرفی پایین آن، پشتیبانی از پروتکل I2C جهت ارتباط با ماژول فراصوت، ابعاد کوچک و همچنین پروتکل UART برای ارتباط با سایر میکروکنتلرها می‌باشد.[۲۸]

در شکل ۳-۸ نقشه‌ی پایه های این حسگر نشان داده شد. مطابق این شکل دوپایه‌ی SDA و SCL باید به ترتیب به پایه‌های SDA و SCL انتخاب شده در Mega8 متصل شوند و همچنین تغذیه‌ی ماژول نیز باید به صورت مستقل به پایه‌های +5v Power و Gnd متصل شود. در این ماژول ثبات‌های کنترلی در نظر گرفته شده‌اند که با نوشتن مقدار خاصی در آنها می‌توان نوع اطلاعات بازگشتی را تنظیم کرد و سایر تنظیمات این نوع ماژول‌ها را اعمال نمود. شکل ۳-۱۰ این جدول تنظیمات را برای SRF08 نشان می‌دهد: [۲۷]

| Command | | Action |
|---|---|---|
| Decimal | Hex | |
| 80 | 0x50 | Ranging Mode - Result in inches |
| 81 | 0x51 | Ranging Mode - Result in centimeters |
| 82 | 0x52 | Ranging Mode - Result in micro-seconds |
| | | |
| 83 | 0x53 | ANN Mode - Result in inches |
| 84 | 0x54 | ANN Mode - Result in centimeters |
| 85 | 0x55 | ANN Mode - Result in micro-seconds |
| | | |

*شکل ۳- ۱۰ جدول دستورات معتبر برای ارتباط با ماژول فراصوت SRF08*

طبق اعلام کارخانه‌ی سازنده‌ی این ماژول در صورت نوشتن مقدار 0X51 در رجیستر به آدرس 0x00 ماژول با یک تاخیر حدود ۷۰ میلی‌ثانیه‌ای خروجی‌ای بر حسب سانتی‌متر می‌دهد. همچنین طبق استاندارد اعلام شده توسط کارخانه سازند در صورت ارسال اطلاعات به ترتیب زیر این امر امکان پذیر است:

۱. آدرس خود ماژول
۲. آدرس رجیستر
۳. داده‌ای که می‌خواهیم روی رجیستر بریزیم و مشخص کننده‌ی نوع داده‌ی بازگشتی است

بنابراین با دانستن این ترتیب به نحوی که در شکل ۳-۱۱ نشان داده شده است، بخشی از عمل دریافت اطلاعات از ماژول را انجام می‌دهیم که همان اعلام نوع درخواست است.

```

#include <mega8.h>
#include <delay.h>
#asm
   .equ __i2c_port=0x18 ;PORTB
   .equ __sda_bit=0
   .equ __scl_bit=1
#endasm
#include <i2c.h>

int Range1;
int Range2;

```

*شکل ۳- ۱۱ تنظیمات I2C و تعریف متغییر ها برای راه اندازی SRF08*

مطابق شـکل ۳-۱۱ خطهای ۲۷، ۲۸ و ۲۹پروتکل I2C روی پورت B میکروکنتلر Mega 8 تعریف شد. به نحوی که بیت صفر پورت B به عنوان sda و بیت یک آن پورت به عنوان scl معرفی شد.

با اسـتفاده از تابع srf08_write( ) عمل اندازه‌گیری فاصـله توسـط حسـگر قابل انجام است. شکل ۳-۱۲ پیاده‌سازی این تابع را نشان می‌دهد. در ورودی‌های این تابع بخش device_addr باید آدرس ماژول که در حالت اولیه همان 0x00 اسـت قرار داد، در بخش register_addr آدرس رجیســـتری کـه می‌خواهیم روی آن داده را بریزیم می‌آیـد کـه ما در اینجا چون می‌خواهیم کار با فاصـــله را انجام دهیم طبق گفته‌ی کارخانه‌ی سـازنده با رجیستر 0x00 کار می‌کنیم. سپس در بخش data همان‌طور که در شکل ۳-۱۰ مشاهده شد، مقدار 0x51 را قرار می‌دهیم چراکه می‌خواهیم اطلاعات ما بر حسب سانتی‌متر باشد.

```
void srf08_write(char device_addr, char register_addr, char data)
{
        i2c_start();
        i2c_write(device_addr);
        i2c_write(register_addr);
        i2c_write(data);
        i2c_stop();
        delay_ms(100);
}
```

*شکل ۳- ۱۲ پیاده سازی تابع srf08_write، این تابع دستور کنترلی مورد نظر را به ماژول می‌فرستد.*

اگر طبق ترتیب زیر اطلاعات جدیدی را برای ماژول بفرستیم می‌توانیم اطلاعات محاسبه شده در ماژول را بخوانیم.

۱. ارسال آدرس ماژول
۲. ارسال آدرس رجیستر خواندن که در اینجا 0x02 است.
۳. ارسال آدرس ماژول +۱
۴. دوبار خواندن خط داده که اولین خواندن برای خواندن بایت پرارزش بالایی فاصله‌ی محاسبه شده است و دومین خواندن برای خواندن بایت پایینی یا کم ارزش‌تر فاصله‌ی محاسبه شده است.

شـکل ۳-۱۳ پیاده سازی تابع خواندن اطلاعات از ماژول را با استفاده از الگوریتم بالا نشان می‌دهد. خط ۵۹ در این شکل، دو بایت کم‌ارزش و پرارزش اطلاعات دریافتی از ماژول را کنار هم قرار می‌دهد. متغییر Range که توسـط تابع برگشـت داده می‌شـود، فاصـله ی اندازه گیری شده توسط حسگر را نشان می دهد.

```
int srf08_read(char device_addr, char register_addr)
{
   int range;
   char vh,vl;
   i2c_start();
   i2c_write(device_addr);
   i2c_write(register_addr);
   i2c_start();
   i2c_write(device_addr + 1);
   delay_ms(100);
   vh = i2c_read(1); //ack
   vl = i2c_read(0); //nack
   i2c_stop();
   range = vh * 256 + vl;
   return range;
}
```

*شکل ۳- ۱۳ پیاده سازی تابع خواندن اطلاعات از حسگر فراصوت SRF08*

شـکل ۳-۱۴ مثالی از کاربرد دو تابع ()srf08_read و ()srf08_write را نشـان می‌دهد. در خطوط ۲۰۶ و ۲۰۷، ارسـال فرمان و دریافت اطلاعات از یک حسگر فراصوت که در آدرس 0x00 قرار دارد انجام می‌شود و نتیجه در متغییر Range1 ذخیره می‌شود. در خطوط ۲۰۹ و ۲۱۰ همین ارسـال و دریافت برای یک حسگر فراصوت SRF08 که در آدرس 0xF8 قرار دارد، صورت می‌گیرد و فاصله‌ی محاسبه شده در متغییر Range2 ذخیره می شود.

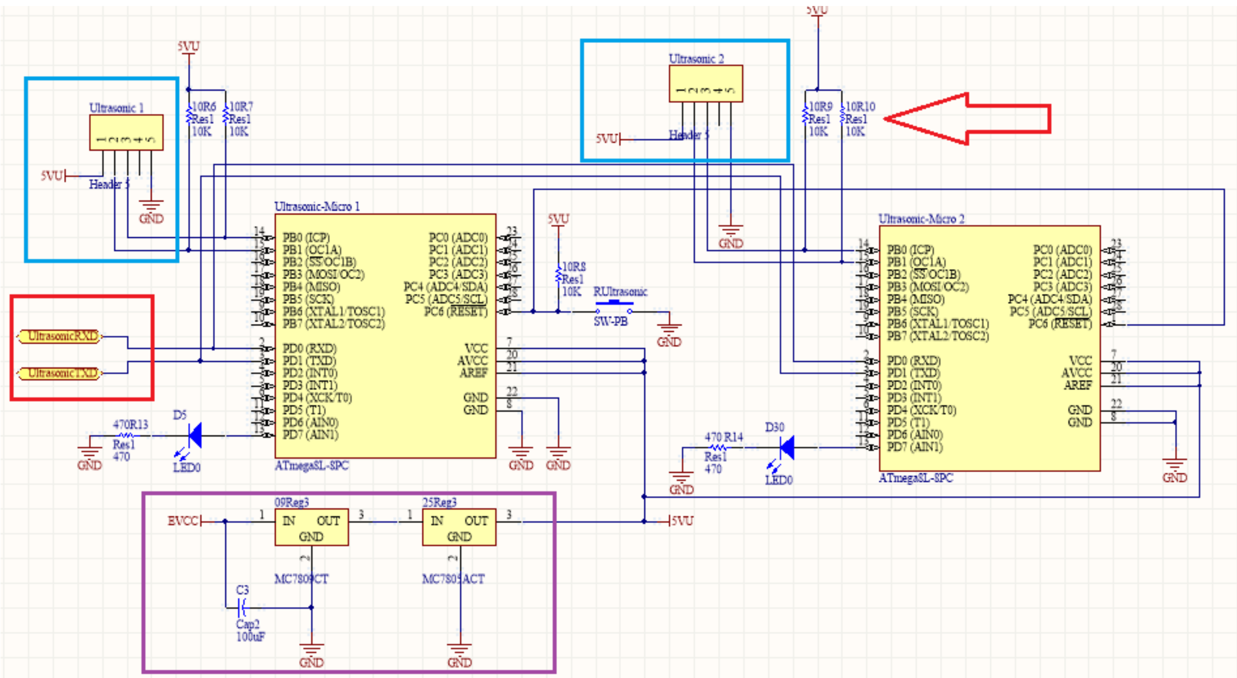
```
202
203
204 while (1)
205         {
206            srf08_write(0x00,0x00,0x51);
207            Range1 = srf08_read(0x00,0x02);
208
209            srf08_write(0xF8,0x00,0x51);
210            Range2 = srf08_read(0xF8,0x02);
```

*شکل ۳- ۱۴ مثالی از ارتباط با دو ماژول SRF08 که در آدرسهای 0x00 و 0xF8 قرار دارند*

شکل ۳-۱۵، شماتیک مدار راه‌انداز حسگرهای فراصوت را نشان می‌دهد. از دو عدد میکروکنترلر ATMega 8 برای راه اندازی هریک از دو حسـگر فراصـوت SRF08 اسـتفاده شـد. مربع بنفش رنگ مدار تغذیه ی میکروکنترلرها و حسـگرها را نشـان می دهد. این مدار که از MC7805ACT و MC7809ACT تشـکیل شده است، اختلاف پتانسیل ۱۲ ولت باتری را ابتدا به ۹ ولت تبدیل می‌کند و سپس ۹ ولت را به ۵ ولت تبدیل می‌نماید. از این اختلاف پتانسیل ۵ ولت جهت تغذیه حسگرها و میکروکنتلر ها استفاده می‌شود[۲۹]. دلیل تبدیل غیر مستقیم ۱۲ ولت به ۵ ولت آن اسـت که در صـورت نوسـان ولتاژ باتری یا تغذیه کننده مدار آسـیبی به MC7805ACT وارد نشـود. در واقع در این حالت MC7809ACT در نقش یک فیوز برای مدار عمل می‌کند. چون MC7805ACT مرز بین مدار حسگر و تغذیه کننده است، حفاظت از این قطعه

اهمیت می‌یابد. زیرا در صورت آسیب دیدن آن ممکن است ولتاژ ۱۲ ولت به طور مستقیم وارد مدار شود و باعث آسیب میکروکنترلرها و حسگرها شود.

هر یک از مربع‌های آبی رنگ نقاط اتصال یکی از حسگرهای فراصوت را به مدار نشان می‌دهد. مربع قرمز رنگ پورتهای ارتباط سریال این مدار با میکروکنترلر مرکزی مدار است. فلش قرمز رنگ به یکی از نکات کلیدی طراحی این مدار اشاره می‌کند. چون حسگرها از پروتکل I2C استفاده می‌کنند، پس باید حتما توسط مقاومت‌های بالاکش (Pull up) به ۵ ولت مثبت متصل شوند[۳۰].




شکل ۳- ۱۵شماتیک مدار حسگرها

### ۳-۴-۳ راه اندازی موتورهای سرو

هر سرو موتور دارای سه عدد سیم می‌باشد. این سیم‌ها غالبا به رنگ‌های سیاه (زمین یا Ground)، قرمز ( ۶+ ولت) و نارنجی (یا زرد) می‌باشند. سیم نارنجی، پالس کنترل کننده‌ی سرو موتور است و از پروتکل مدلاسیون عرض پالس یا PWM پشتیبانی می‌کند. مدار داخلی سرو موتور، در هر لحظه duty cycle (مدت زمانی که پالس در سطح بالا قرار دارد) پالس کنترلی را با موقعیت کنونی خود مقایسه می‌کند و متناسب با آن موقعیت بعدی خود را تنظیم می‌کند. فرکانس کاری پالس کنترل کننده باید ۵۰ کیلوهرتز باشد. با ۵۰ کیلوهرتز، عرض پالس کنترلی ۲۰ میلی‌ثانیه است. سرو برای باقی ماندن در یک زاویه خاص باید مرتبا پالس کنترلی را تا زمان تغییر زاویه و رفتن به زاویه جدید دریافت نماید.

زمانی که duty cycle پالس کنترلی یک میلی‌ثانیه باشد، سرو در منتی الیه سمت چپ خود در زاویه ۱۸۰ درجه نسبت به صفر خود می‌ایستد. اگر duty cycle پالس کنترلی ۲ میلی ثانیه باشد، سرو در منتهی الیه سمت راست خود در زاویه صفر درجه می‌ایستد. اگر مقادیر duty cycle بین یک تا دو میلی ثانیه تغییر کند، سرو موتور متناسب با آن به صورت خطی در بازه ای بین صفر تا ۱۸۰ درجه تغییر می کند. برای مثال اگر duty cycle برابر با ۱٫۵ میلی ثانیه باشد، سرو موتور در زاویه ۹۰ درجه خواهد ایستاد.

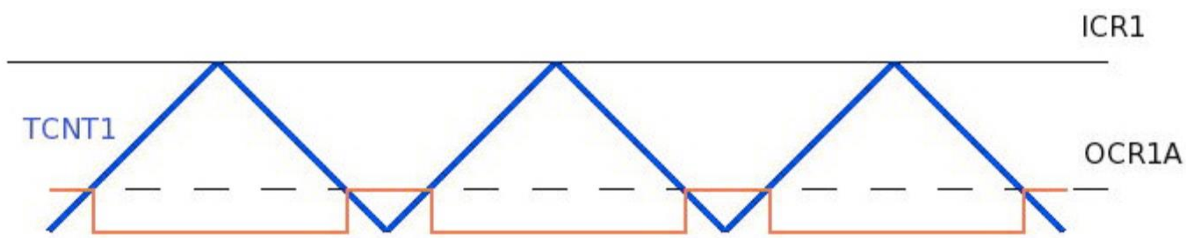
برای تولید پالســـی با عرض ۲۰ میلی‌ثانیه و duty cycle بین ۱ تا ۲ میلی‌ثانیه از پیکربندی تایمر ATMega 32 در حالت Phase and Frequency Correct Mode استفاده شد. در این حالت تایمر از صفر شروع می‌کند و تا یک عدد مشخص و قابل تنظیم به نام ICR می‌شمارد و پس از رســیدن به آن دوباره به ســمت صـفر شــمارش معکوس می‌کند. این روال مرتبا ادامه پیدا می‌کند. در این حالت دو عدد فرکانس پالس ساعت قابل تنظیم در میکروکنتلر وجود دارد: فرکانس پالس ساعت تایمر و فرکانس پالس ساعت میکروکنترلر.[۱۱]

فرکانس پالس ساعت میکرو مقداری مشخص و از پیش تعیین شده است. در راه اندازی موتور سرو، این مقدار ۸ مگاهرتز در نظر گرفته شد. فرکانس پالس ساعت تایمر، خارج قسمت صحیحی از تقسیم فرکانس پالس ساعت میکروکنتلر بر بعضی توان هایی از ۲ می‌باشند. مقسوم علیه این تقســیم به اختصــار N نامیده می‌شــود و می‌تواند مقادیر ۱، ۲، ۴، ۸، ۶۴، ۲۵۶، ۱۰۲۴ را در برگیرد. برای تولید فرکانس دلخواه، از معادله‌ی ۳-۱ استفاده می شود:[۱۱]

$$fo = \frac{fm}{2*N*ICR}$$ معادله ۳-۱

در معادله۳-۱، fo فرکانس مورد نظر در خروجی PWM میکروکنتلر است. fm فرکانس پالس ساعت میکروکنتلر می‌باشد. N مقسوم‌علیه تقسیم برای انتخاب فرکانس پالس ساعت تایمر میکروکنتلر می‌باشد. ICR نیز قبلا توضیح داده شد.

در راه اندازی ســرو موتور در این پروژه، با در نظر گرفتن فرکانس پالس ســاعت ۸ مگاهرتز برای میکروکنتلر و مقدار N = 8 ، فرکانس پالس ســاعت تایمر یک مگاهرتز بدســت آمد. با استفاده از این مقادیر و با جایگذاری آنها در معادله ۳-۱ مقدار ICR = 20000 بدست می‌آید. با معلوم شــدن ICR و با مقدار ۲۰۰۰۰، یک پالس مربعی با فرکانس ۵۰ هرتز در خروجی PWM میکروکنتلر تولید می‌شــود. اما هنوز این پالس قدرت کنترل سرو موتور را نخواهد داشت. زیرا duty cycle آن ممکن است در حوزه‌ی مقادیر معتبر برای کنترل سرو نباشد. برای تنظیم duty cycle شکل موج خروجی PWM از ثبات قابل مقداردهی OCR استفاده می‌شود. شکل ۳-۱۶ نحوه ی دقیق کار سیستم تایمر یک میکروکنتلر را برای تولید پالس PWM در خروجی آن نشان می دهد:




*شکل ۳- ۱۶ شکل موج های تایمر میکروکنتلر برای تولید PWM، پالس نارنجی رنگ، پالس تولید شده در خروجی PWM میکروکنتلر است. خط نقطه‌چین، OCR1A و خط آبی مقدار TCNT را نشان می‌دهند.*

مطابق با این شـکل، ثبات TCNT که نقش شمارنده ی تایمر را دارد، با شروع کار تایمر از صفر شروع کرده و تا مقدار ICR می‌شمارد. وقتی TCNT به مقدار ICR رسـید، به صـورت نزولی شـروع به شمارش می کند تا به صفر برسد. این روال مرتبا ادامه می‌یابد. در شکل ۳-۱۶ پالس مثلثی شـکل آبی رنگ TCNT و پالس نارنجی رنگ مربعی شکل PWM را نشان می‌دهد. مطابق این شکل، با تغییر OCR می توان مدت زمان یک بودن ســیگنال PWM یا به عبارتی دیگر duty cycle آن را تغییر داد. بنابراین، برای کنترل موتور ســرو کافی اســت یک بار ICR مقداردهی شود و پس از آن با تغییر OCR موتور حرکت می کند.

برای راه اندازی سـرو در این پروژه از تایمر شـماره یک میکروکنتلر ATMega 32 اسـتفاده شد. این تایمر قادر است دو پالس خروجی PWM بـه ترتیـب در پایه‌های OC1A و OC1B تولید کند. مقادیر OCR برای کنترل این دو پالس به ترتیب OCR1A و OCR1B می‌باشـــند. تایمر یک، تنها دارای یک مقدار ICR به نام ICR1 می‌باشد. شکل ۳-۱۷ کدهای لازم برای پیکربندی تایمر یک را نشان می‌دهد.

```
47 // Timer/Counter 1 initialization
48 // Clock source: System Clock
49 // Clock value: 1000.000 kHz
50 // Mode: Ph. & fr. cor. PWM top=ICR1
51 // OC1A output: Non-Inv.
52 // OC1B output: Non-Inv.
53 // Noise Canceler: Off
54 // Input Capture on Falling Edge
55 // Timer 1 Overflow Interrupt: Off
56 // Input Capture Interrupt: Off
57 // Compare A Match Interrupt: Off
58 // Compare B Match Interrupt: Off
59 TCCR1A=0xA0;
60 TCCR1B=0x12;
61 TCNT1H=0x00;
62 TCNT1L=0x00;
63 ICR1H=0x4E;
64 ICR1L=0x20;
65 OCR1AH=0x00;
66 OCR1AL=0x00;
67 OCR1BH=0x00;
68 OCR1BL=0x00;
```

*شکل ۳- ۱۷ پیکربندی تایمر شماره یک برای راه اندازی سرو، این تایمر با فرکانس ۱ مگاهرتز (۱۰۰۰ کیلوهرتز) راه‌اندازی شده است.*

این مجموعـه کـد، بـا فرض فرکانس ۸ مگاهرتز برای میکرو کنتلر، تایمر یک را پیکر بندی می‌کند. خط های ۵۹ و۶۰، تایمر یک را با فرکانس یک مگاهرتز (N=8 ) و در حالت Phase and Frequency Correct Mode تنظیم می‌کنند. این دو خط دستور، همچنین باعث می‌شوند مقدار حد شـمارش TCNT1 برابر با ICR1 تنظیم شـود. خط‌های ۶۱و ۶۲ مقدار اولیه TCNT1 را برابر صـفر قرار می‌دهند. این موضـوع باعث می‌شـود TCNT1 از صـفر شـروع به شـمارش کند. خط های ۶۳ و ۶۴ مقدار ICR1 را برابر ۲۰۰۰۰ قرار می‌دهند. نهایتا خط های ۶۵ تا ۶۸، مقادیر اولیه OCR1A و OCR1B را صـفر تنظیم می‌کنند. با این تنظیمات زاویه صـفر درجه سـروها در OCR1A(or B) = 400 اتفاق می‌افتد. شکل ۳-۱۸ برنامه کنترل سرو را نشان می‌دهد.

```
110 OCR1A = 400;
111 OCR1B = 400;
112 while (1)
113         {
114             home:
115             a = getchar();
116             if( a == 'S')
117             {
118 
119               OCR1A = OCR1A + 200;
120               PORTC.0 = 1;
121               if(OCR1A >= 1400)
122                               OCR1A = 400;
123 
124 
125             }
126 
127             if( a == 's')
128             {
129               OCR1B = OCR1B + 200;
130               PORTC.1 = 1;
131               if(OCR1B >= 1400)
132                               OCR1B = 400;
133 
134             }
```

*شکل ۳- ۱۸ برنامه کنترل سرو موتورها*

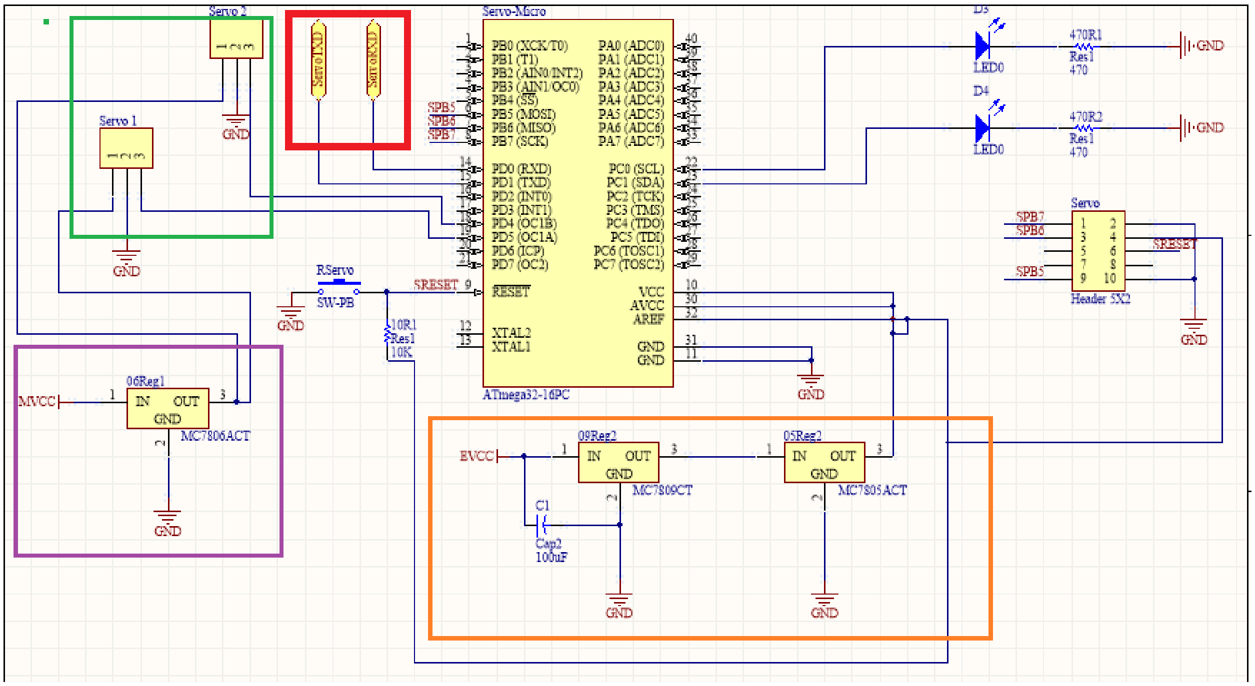
در شــکل ۳-۱۸، در خط‌های ۱۱۰ و ۱۱۱ ابتدا زاویه‌ی صـفر درجه‌ی سـروها تنظیم می‌شود. سپس برنامه وارد یک حلقه‌ی بی‌نهایت while(1) می‌شود. در این حلقه، برنامه منتظر دریافت یک کاراکتر کنترلی از میکروکنترلر مرکزی می‌باشد. اگر کاراکتر ‘S’ بود سرو شماره یک که به پایه‌ی OC1A متصـل اسـت ۳۶ درجه جابجا می‌شـود. این کار با اضـافه کردن ۲۰۰ واحد به مقدار OCR1A (خط ۱۱۹) انجام می‌شود. پس از افزایش چنانچه سرو موتور به زاویه ای فراتر از ۱۸۰ درجه رسیده بود، بلافاصله به صفر بازگردانده می‌شود (خطوط ۱۲۱ و ۱۲۲). همین اتفاق برای سرو موتور شماره دو با کاراکتر کنترلی ‘s’ قابل انجام است. در این پروژه گام‌های سرو موتور ۳۶ درجه در نظر گرفته شد. سرو موتور یک ۱۸۰ درجه کامل را در طی ۵ گام (۵ بار دریافت کارکتر کنترلی) طی می‌کند. مادامی که ســرو کاراکتر کنترلی دریافت نمی‌کند، در زاویه قبلی خود می‌ماند. توجه شـود سـرو موتور برای ماندن در یک زاویه جریان می‌کشد و توان مصرف می‌کند. برای کنترل سرو موتور با تعداد گام دلخواه از معادله ۳-۲ استفاده می‌شود:

$$c = \frac{1000}{s}$$ معادله ۳-۲

در معادله ۳-۲، c مقداری است که باید با OCR1 در شکل ۳-۱۸ جمع شود تا سرو حرکت کند. s تعداد گامهایی است که می‌خواهیم سرو یک ۱۸۰ درجه کامل را با آن طی کند. توجه شود این فرمول فقط برای برنامه نوشته شده در شکل ۳-۱۸ صادق می‌باشد.

شکل ۳-۱۹، شماتیک مدار کنترل کننده سرو موتور را نشان می‌دهد. از یک عدد میکروکنترلر ATMega 32 جهت کنترل دوعدد سرو موتور اســتفاده شده است. مربع سبز رنگ، محل اتصال هر یک از سرو موتورها را به مدار نشان می‌دهد. این سرو موتورها با نام‌های Servo 1 و Servo 2 در مدار مشـخص شـده‌اند و به ترتیب به پایه‌های شماره ۱۹ (OC1A) و شماره ۱۸ (OC1B) میکروکنتلر متصل شده‌اند. مربع نارنجی رنگ مدار تغذیه‌ی میکروکنتلر را نشـان می‌دهد. شرح این مدار دقیقا مانند مدار تغذیه حسگر فراصوت است. مربع بنفش رنگ مدار تغذیه سرو موتورها را نشـان می‌دهد. از MC7806ACT برای تبدیل اختلاف پتانســیل ۱۲ ولت باتری به اختلاف پتانســیل ۶ ولت مورد نیاز ســرو موتورها اســتفاده شــده اســت.[۲۹] توجه شود که چون در دو سر باتری تغذیه کننده موتورها از یک عدد MC7812ACT برای متعادل ساختن اختلاف

پتانسـیل و حفاظت از نوسـان اسـتفاده شـده اسـت، در مدار تغذیه کننده سـرو موتور، تبدیل غیرمستقیم اختلاف پتانسیل(همانند حسگرهای فراصوت) صورت نگرفت.



*شکل ۳- ۱۹ مدار کنترل کننده سرو موتور*

## ۳-۴-۴ راه اندازی انکودرهای DSPR

در شـکل ۳-۳ پیاده سـازی مکانیکی انکودرهای روبات نشـان داده شـدند. در آن شـکل ملاحظه شـد که در انتهای موتورهای روبات پره‌هایی دندانه‌دار قرار دارد. با قرار دادن یک شـمارنده‌ی نوری در بین پره‌ها، تعداد دورهایی ک موتور چرخیده است شمرده می‌شود. همچنین ذکر شـد که با توجه به سـاختار مکانیکی روبات، با قرار دادن یک انکودر بر روی چرخ راسـتای x و یک انکودر بر روی چرخ راسـتای y، در هر لحظه به طور دقیق مختصات روبات را در دو بعد خواهیم داشت.

با چرخش پره‌ها در میان شـمارنده‌ی نوری، شـمارنده پالس های مربع شـکل منظمی را تولید می‌کند. تعداد این پالس‌ها متناسـب با تعداد دوری اسـت که چرخ چرخیده اسـت و همچنین برابر با مسـافتی اسـت که چرخ طی کرده است. برای شمارش این پالس‌ها، از یک عدد میکروکنتلر ATMega 32 به عنوان پردازنده‌ی بخش انکودر اسـتفاده شـد. تایمر صـفر میکروکنتلر، به انکودر راسـتای x روبات و تایمر یک میکروکنتلر به انکودر راسـتای y روبات متصـل شـد. زیرا تایمر یک ۱۶ بیتی اسـت و فرض بر این بود که مسـافتی که روبات در راستای y به عنوان جهت اصلی خود طی می‌کند، بیشتر است. بنابراین بهتر است از تایمری استفاده شود که ظرفیت بالاتری دارد.

شـکل ۳-۲۰ نحوه ی پیکربندی تایمرهای صـفر و یک را برای شـمارش پالس‌های انکودر نشان می‌دهد. همانطور که ملاحظه می‌شود فقط ثبات‌های TCCR این تایمرها مقداردهی شـده اند و مقدار بقیه ثبات‌ها صفر است. مقداردهی ثبات‌های TCCR باعث شده است که هر دو تایمر در حالت شمارنده فعال شوند. تایمر صفر تعداد لبه‌های پایین رونده‌ی پالسی را که به پایه ی T0 میکروکنتلر متصل شده است می‌شمارد. تایمر یک تعداد لبه‌های بالارونده‌ی پالسی را که به پایه ی T1 میکروکنتلر متصل شده است می‌شمارد.

```
84  // Timer/Counter 0 initialization
85  // Clock source: T0 pin Falling Edge
86  // Mode: Normal top=FFh
87  // OC0 output: Disconnected
88  TCCR0=0x06;
89  TCNT0=0x00;
90  OCR0=0x00;
91
92  // Timer/Counter 1 initialization
93  // Clock source: T1 pin Rising Edge
94  // Mode: Normal top=FFFFh
95  // OC1A output: Discon.
96  // OC1B output: Discon.
97  // Noise Canceler: Off
98  // Input Capture on Falling Edge
99  // Timer 1 Overflow Interrupt: On
100 // Input Capture Interrupt: Off
101 // Compare A Match Interrupt: Off
102 // Compare B Match Interrupt: Off
103 TCCR1A=0x00;
104 TCCR1B=0x06;
105 TCNT1H=0x00;
106 TCNT1L=0x00;
107 ICR1H=0x00;
108 ICR1L=0x00;
109 OCR1AH=0x00;
110 OCR1AL=0x00;
111 OCR1BH=0x00;
112 OCR1BL=0x00;
```

*شکل ۳- ۲۰ پیکربندی تایمرهای صفر و یک میکروکنتلر برای شمارش پالس های انکودر، هر دو تایمر در حالت شمارنده پیکربندی شده‌اند.*

در خط‌های شماره ۸۶ و ۹۴ شکل ۳-۲۰، برای تایمرهای صفر و یک، یک حد مشخص شده است. این حد برای تایمر ۸ بیتی صفر، ۲۵۵ و برای تایمر ۱۶ بیتی یک، ۶۵۵۳۵ می‌باشد. زمانی که تعداد پالس های شمرده شده توسط هر یک از تایمرها به حد مربوط آن برسد، آن تایمر یک وقفه سرریز را فراخوانی می‌کند. شکل ۳-۲۱ روال وقفه سرریز تایمر صفر و یک را نشان می‌دهد. همانطور که ملاحظه می‌شود در این روال‌ها به ترتیب یک متغیر به نام countx و county یک واحد افزایش می‌یابد. همچنین یک پین میکروکنتلر ولتاژ سطح بالا می‌گیرد. این پین به یک دیود نورانی متصل شده است که نشان می‌دهد تایمر سرریز شده است. متغییر های countx و county تعداد دفعاتی را که هر یک از تایمرها سرریز شده اند نشان می دهد.

شکل ۳-۲۲ نحوه ی گزارش‌دهی مختصات روبات به میکروکنتلر اصلی را نشان می‌دهد. میکروکنتلر بخش انکودر مرتبا در یک حلقه‌ی بی نهایت while(1)، انتظار دریافت یک کاراکتر را می‌کشد. اگر کاراکتر دریافتی 'x' باشد، کد خط های ۱۶۴ تا ۱۷۴ اجرا می‌شود. در این کدها ابتدا محتوی countx در ۲۵۵ ضرب می‌شود. زیرا هر بار که تایمر صفر به ۲۵۵ می‌رسید صفر می شد و یک واحد به countx اضافه می‌کرد. برای آنکه تعداد کل پالس‌هایی که این تایمر شمرده است محاسبه شود، این عمل ضرب صورت می‌گیرد. در خط شماره ۱۶۵، مقدار این حاصلضرب با مقدار کنونی تایمر صفر (TCNT0) جمع می‌شود. زیرا ممکن است در لحظه‌ی گزارش هنوز تایمر سرریز نشده باشد و مقداری از شمارش ها درون خود شمارنده باشد. بدین وسیله تمامی پالس‌هایی که تا کنون توسط این تایمر شمرده شده است در متغییر stepx قرار می‌گیرد. طبق آزمایش‌های تجربی انجام شده برای روبات، هر ۵۶ پالس شمارش شده معادل یک سانتی‌متر مسافت پیموده شده توسط روبات است. بنابراین در خط ۱۶۶ مقدار مسافت پیموده شده توسط روبات در راستای x بر حسب سانتی‌متر محاسبه می‌شود. توجه شود که عدد ۵۶ با فرض چرخ‌های کنونی روبات می‌باشد که قطر ۵ سانتی‌متر دارند. در صورت تعویض چرخ ها این عدد متناسب با قطر چرخ جدید تغییر

می‌کند. در خط شماره ۱۶۷ مقدار مسافت پیموده شده در راستای محور x از طریق ارتباط سریال به میکروکنتلر مرکزی ارسال می‌شود. سپس مقدار تمامی متغیرهایی که در شمارش مسافت پیموده شده در راستای x دخیل بودند صفر می‌شود. زیرا در طراحی به دلیل مسایل حفاظت از نویز، فرض بر این شد که میکروکنتلر مرکزی حساب مسافت‌های پیموده شده و محاسبه‌ی جابجایی را بر عهده بگیرد. بدین ترتیب چنانچه نویز یا تداخلی در شـمارش انکودرها وارد شـود، فقط مقادیر مربط به یک بار گزارش دچار مشـکل می‌شود. اگر کاراکتر دریافتی 'y' باشد، کدهای خط های ۱۷۷ تا ۱۸۶ انجام می‌گیرند. عملکرد این کدها دقیقا مانند کدهای پاراگراف قبلی است با این تفاوت که به دلیل ۱۶ بیتی بودن تایمر یک و سرریز شدن آن در عدد ۶۵۳۵، متغییر county در عدد ۶۵۳۵ ضرب می‌شود.

```
// Timer 0 overflow interrupt service routine
interrupt [TIM0_OVF] void timer0_ovf_isr(void)
{
// Place your code here
  countx = countx + 1;
  PORTC.0 = 1;

}

// Timer 1 overflow interrupt service routine
interrupt [TIM1_OVF] void timer1_ovf_isr(void)
{
// Place your code here
   county = county + 1;
   PORTC.1 = 1;
}
```

*شکل ۳- ۲۱ روال های سرویس دهی به وقفه های سرریز تایمر های صفر و یک، پس از هربار سر ریز شدن تایمر صفر یا تایمر یک متغییرهای countx یا county به ترتیب یک واحد افزایش می‌یابند.*

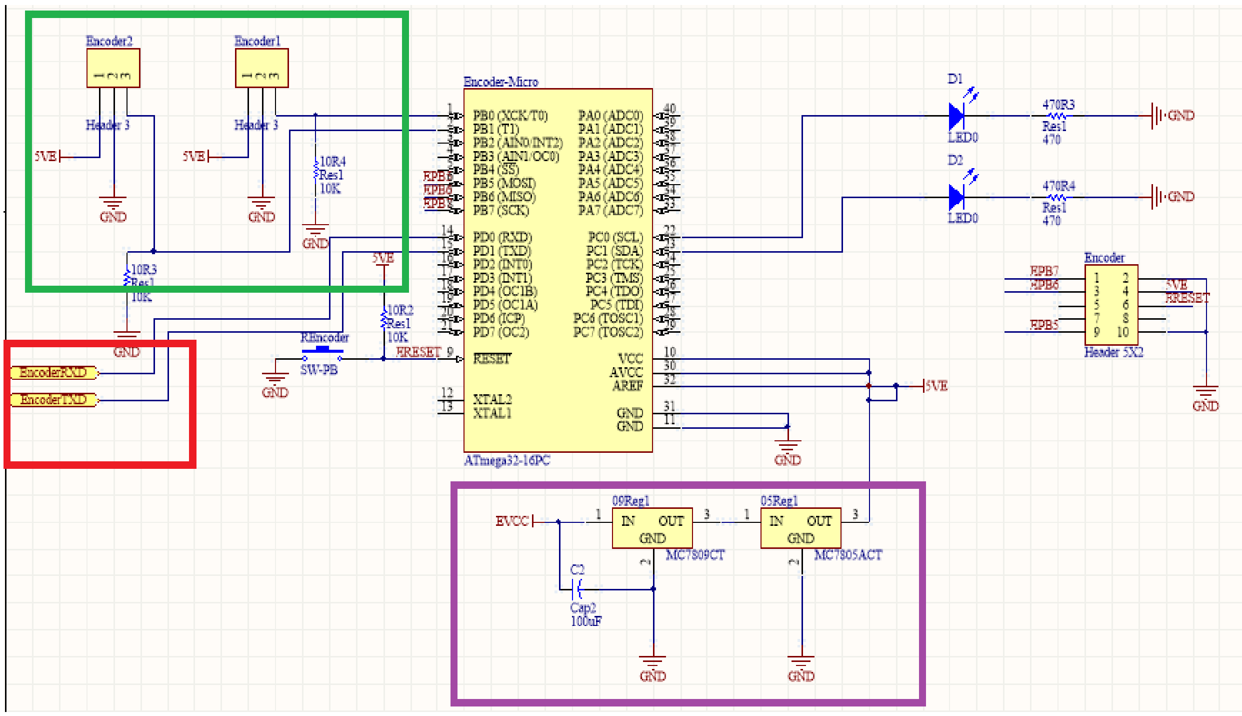
```
while (1)
      {
      // Place your code here
      home:
      PORTC = 0;
      PORTB.0 = 1;
      PORTB.1 = 1;
      a = getchar();
      if( a == 'x')
          {
           stepx = countx * 255 + TCNT0;
           stepx = stepx / 56;
           putchar(stepx);
           TCNT0 = 0;
           countx = 0;
           stepx = 0;
//            sprintf(b,"x = %d cm",stepx);
//            puts(b);

          }

      if( a == 'y')
      {
          stepy = county * 65535 + TCNT1;
          stepy = stepy / 56;
          putchar(stepy);
          TCNT1 = 0;
          stepy = 0;
          county =0;
//          sprintf(b, "y = %d cm", stepy);
//          puts(b);
      }
```

*شکل ۳-۲۲ برنامه‌ی انکودرهای روبات، این برنامه منتظر کاراکتر کنترلی می‌شود، پس از دریافت کاراکتر کنترلی، مقدار انکودر مربوط به آن را ارسال می‌نماید، سپس متغییرهای انکودرها reset می‌شوند.*

شکل ۳-۲۲ شماتیک مدار انکودر را نشان می‌دهد. از یک میکروکنترلر ATMega 32 به عنوان پردازنده‌ی اصلی واحد انکودر استفاده شده است. کادر سبز رنگ محل اتصال انکودرها را به مدار روبات نشان می‌دهد. انکودر راستای x (Encoder 1) به پایه‌ی شماره یک میکروکنترلر(T0) و انکودر راستای y (Encoder 2)، به پایه‌ی شماره دو میکروکنترلر (T1) متصل شده است. کادر بنفش مدار تغذیه میکروکنترلر را نشان می‌دهد. در مورد این مدار تغذیه به طور کامل در بخش ۳-۴-۲ بحث شد. کادر قرمز رنگ نیز نقاط اتصال مدار انکودر را به میکروکنترلر اصلی از طریق ارتباط سریال نشان می دهد.



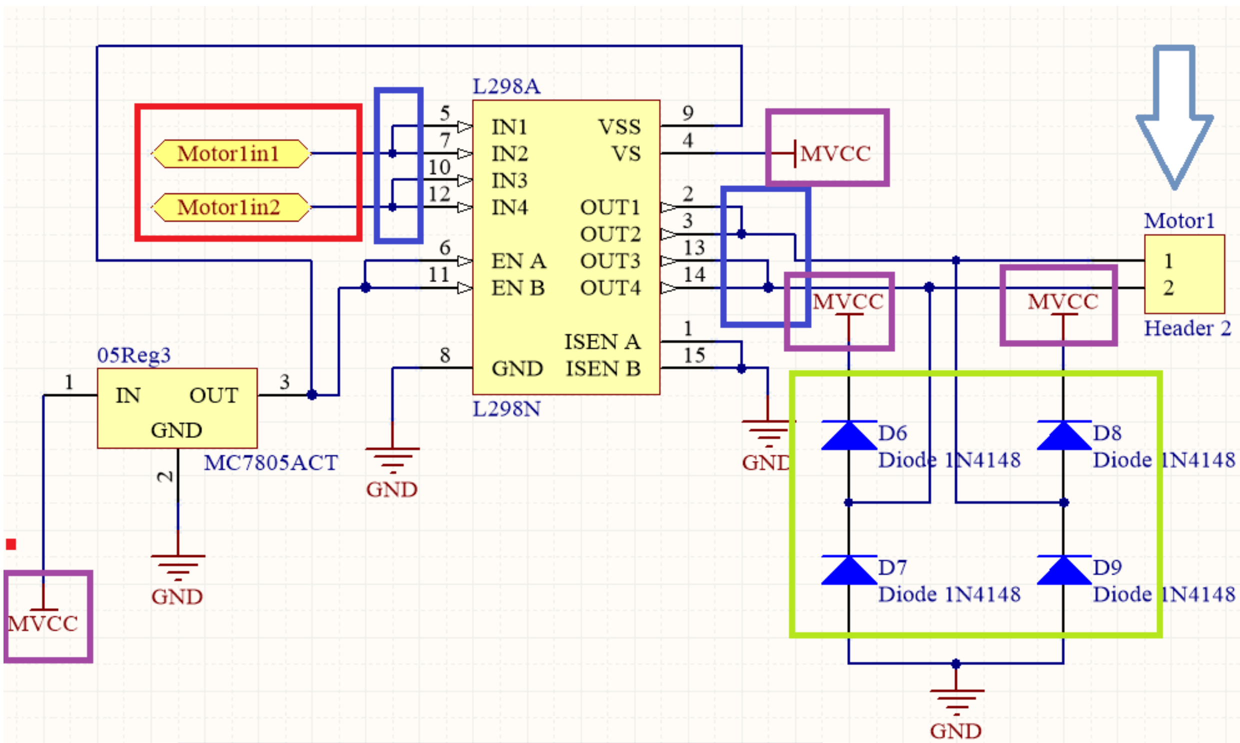

*شکل ۳-۲۳ شماتیک مدار انکودر*

### ۳-۴-۵ راه اندازی درایور موتورهای حرکتی DSPR

همانطور که در بخش ۲-۶-۶مطرح شـد، موتورهای حرکتی روبات موتورهای DC می‌باشند. همچنین در بخش۳-۲-۲مشخص شد که نوع موتورهای روبات L012CR2342 سـاخت شرکت FAULHABER است. این موتورها با اختلاف پتانسیل ۱۲ ولت و جریان عملیاتی متوسط ۱۰۰ میلی آمپر راه‌اندازی می‌شـوند. کنترل این موتورها توسـط میکروکنترلر اصـلی صورت می‌گیرد. اما میکروکنتلر به طور مستقیم قادر به راه اندازی این موتورها نیسـت. زیرا در حالت عملیاتی ATMega 32 تنها می تواند اختلاف پتانسـیل ۵ ولت را با جریان حداکثر ۳۰ میلی‌آمپر در خروجی خود ایجاد نماید.[۱۱] این اختلاف پتانسـیل و جریان، برای راه اندازی یک موتور DC با شـرایط ذکر شده کافی نمی‌باشد. برای تامین اختلاف پتانسیل و جریان مورد نیاز از یک مدار مجتمع درایور استفاده می‌شود. پس از بررسی بر روی درایورهای مختلف، آی سی درایور L298 به دلیل محبوبیت در بازار داخلی، قیمت مناسـب، توان سوئیچ ولتاژ بالا (تا ۴۶ ولت به صورت عملیاتی) و تحمل جریان بالا تا ۲ آمپر به صورت عملیاتی (۴ آمپر به صورت اسمی) انتخاب شد.[۳۱]

شکل شماره ۳-۲۴، شماتیک درایور یکی از موتورهای حرکتی را نشان می‌دهد. شماتیک کلی برای همه‌ی موتورها، شامل ۴ بار تکثیر از این شماتیک با اعمال اندکی جزئیات می‌باشد. در این شکل کادر قرمز رنگ نقطه‌ی اتصال مدار درایور موتور شماره ۱ را به میکروکنترلر اصلی نشـان می‌دهد. این ارتباط صرفا از نوع ورودی/خروجی، یک طرفه و با سطح ولتاژ صفر و ۵ ولت می‌باشد. برای آنکه تحمل توان عملیاتی درایور افزایش یابد ورودی‌ها و خروجی‌های آن دوبه دو با یکدیگر سـری شـده‌اند. این امر موجب ایجاد یک گره در محل‌هایی می‌شـود که با کادر سرمه‌ای نشان داده شده‌اند. در محل این گره‌ها طبق قانون جریان کیرشهف، جریان تقسیم می‌شود و باعث می‌شود هر کانال درایور عملا نصف بار عملیاتی جریان را تحمل نماید.[۳۲] این موضـوع موجب افزایش توان قابل تحمل عملیاتی درایور می‌شـود. کادر سـبز رنگ حفاظت دیودی انجام شـده در مقابل برگشت جریان معکوس القایی از موتورها را نشان می‌دهد. در صورت برگشت جریان معکوس القایی، با وجود دیودها، این جریان به سمت منبع تغذیه هدایت می‌شود و از آسیب زدن به درایور جلوگیری می‌گردد.[۳۳]

جهت جلوگیری از آسیب احتمالی در برابر شوک های الکتریکی که در اثر استارت و توقف موتور ها به مدار تغذیه وارد می شود، تغذیه کل مدار به دوبخش تقسیم شد: یک باتری با نام MVCC (کادرهای بنفش شکل ۳-۲۴) برای تغذیه سرو موتور و موتورهای DC و یک باتری دیگر با نام EVCC که در شکل های ۳-۱۵، ۳-۱۹ و ۳-۲۳ قابل مشاهده است و برای تغذیه بخش الکترونیک روبات و کلیه ی قطعات الکترونیکی آن استفاده می شود. شکل پیکان(فلش) محل اتصال موتور شماره ۱ را به مدار نشان می دهد.

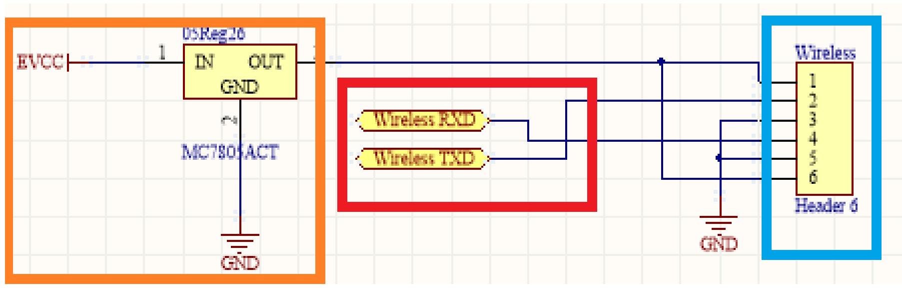



شکل ۳- ۲۴مدار درایور روبات

### ۳-۴-۶ راه‌اندازی ارتباط بی‌سیم روبات

در قسمت۲-۶-۱۳ ذکر شد که روبات جهت پردازش اطلاعات خود، آنها را به یک عامل خارجی که کاملا در خارج از فضای فیزیکی روبات قرار دارد می‌فرستد. جهت ایجاد یک کانال ارتباط بی‌سیم میان روبات و عامل، از یک ماژول فرستنده-گیرنده HM-TR با فرکانس ۹۱۵ مگاهرتز استفاده شد.[۳۴] دلایل انتخاب این ماژول از میان گستره‌ی وسیع ماژول های فرستنده-گیرنده موجود در بازار بومی عبارتند از: پشتیبانی از ارتباط سریال، راه‌اندازی بسیار آسان، قابلیت تنظیم و اختصاص باند فرکانسی، برد مفید عملیاتی نزدیک به ۱۰۰ متر، مصرف جریان پایین ۵۰ میلی‌آمپر، عدم نیاز به پیکربندی، قیمت مناسب و محبوبیت در بازار بومی.

شکل ۳-۲۵، شماتیک مدار ارتباط بی‌سیم روبات را نشان می‌دهد. این شکل بخش عمده‌ای از ادعاهای صورت گرفته در مورد مزایای استفاده از ماژول HM-TR را اثبات می‌کند. زیرا همانطور که ملاحظه می شود، راه‌اندازی این ماژول به حداقل قطعات و مدارات جانبی نیاز دارد. با وجود مصرف توان پایین این ماژول، برای رعایت اصول طراحی ماژولار، برای آن یک سیستم تغذیه مجزا در نظر گرفته شد. کادر نارنجی رنگ این سیستم تغذیه را نشان می‌دهد. کادر آبی رنگ نقطه‌ی اتصال این ماژول به مدار روبات و کادر قرمز رنگ کانال ارتباطی این ماژول از طریق ارتباط سریال با میکروکنتلر اصلی را نشان می‌دهد.



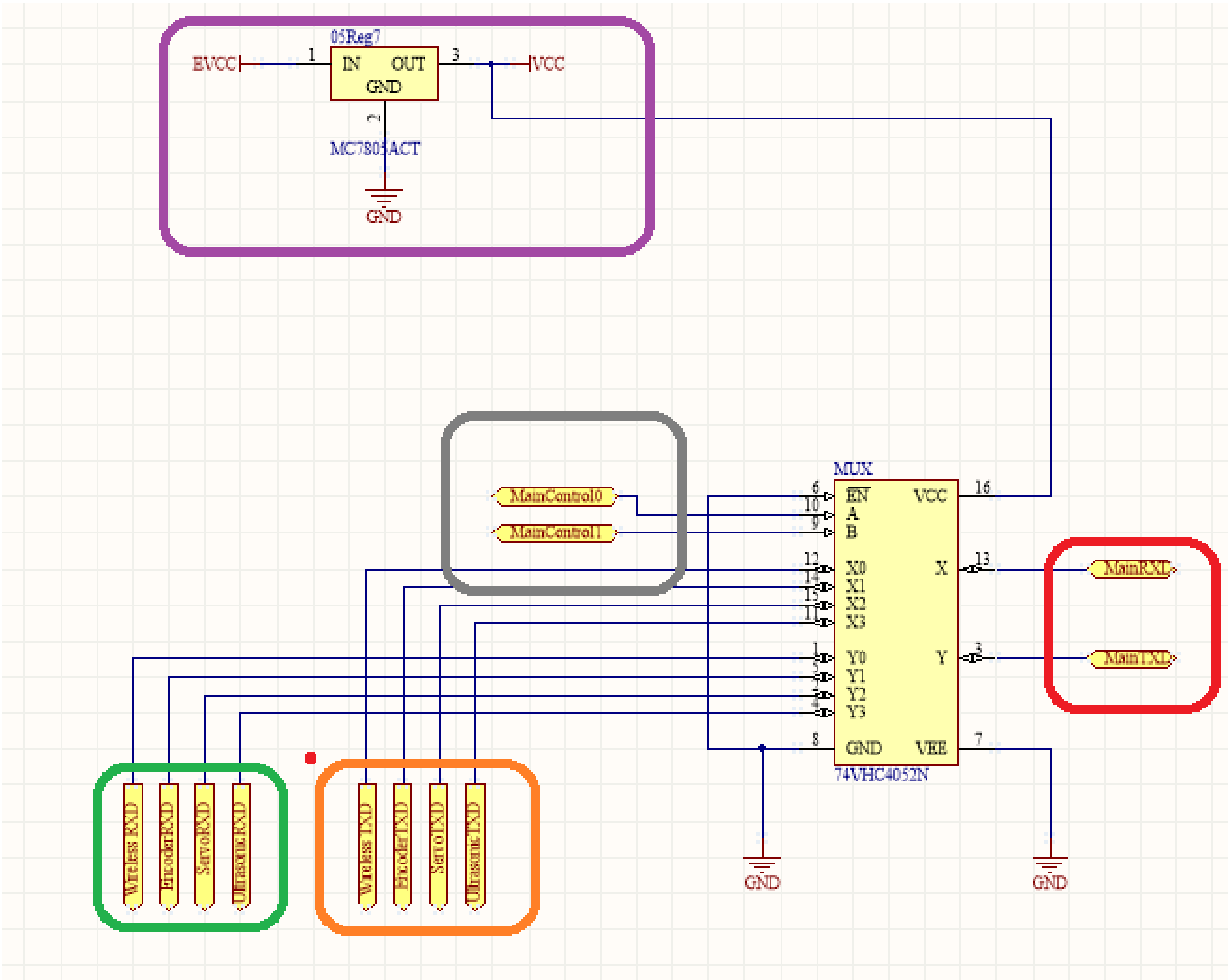

*شکل ۳- ۲۵شماتیک مدار ارتباط بی‌سیم روبات*

به خاطر پشتیبانی میکروکنتلر از سطح ولتاژ TTL، در مدار روبات از یک ماژول HM-TR با فرکانس ۹۱۵ مگاهرتز با مدل TTL و در سمت عامل، از یک ماژول HM-TR با فرکانس ۹۱۵ مگاهرتز با مدل RS232 استفاده شد.[۳۴] بدین ترتیب، عامل کنترل کننده روبات می‌تواند هر سیستم پردازشی باشد که از ارتباط سریال پشتیبانی می‌کند. با این شرط، یکی از مهمترین معیارهای طراحی روبات که ویژگی سکو بودن آن است و در بخش ۳-۴-۱ به صورت کامل بحث شد، برآورده می‌شود. ارتباط با این ماژول بسیار ساده و همانند ارتباط از طریق پورت سریال میکروکنتلر بـا یک میکروکنتلر دیگر می‌باشـد. به طور خاص در این پروژه از دسـتورات ()putchar و ()getchar برای ارتباط با آن اسـتفاده شد.[۳۰]

## ۳-۴-۷ استانداردسازی فرکانس‌ها، نرخ تبادل بیت‌ها و مدیریت دسترسی به کانال ارتباطی

قبل از پرداختن به جزئیات پیاده‌سـازی مدار میکروکنتلر اصـلی، بهتر است یک نکته‌ی مبهم در طراحی مدار روبات مشخص شود. در بخش‌های ۳-۴-۲ تا ۳-۴-۶ مرتبا ذکر شد که بخش‌های حسگر فراصوت، انکودر، فرستنده-گیرنده بی‌سیم و موتور سرو همگی از طریق ارتباط سریال با میکروکنتلر مرکزی(اصلی) در تماس هستند. اما تعداد ارتباط‌های سریالی که یک میکروکنتلر از آن پشتیبانی می‌کند محدود است. در این مورد چون میکروکنتلر مرکزی ATMega32 می‌باشد، فقط از یک ارتباط سریال پشتیبانی می‌کند.[۱۱] این درحالی است که برای ارتباط با بخش‌های مختلف به ۴ عدد ارتباط سریال نیاز است. برای رفع این مشکل و مدیریت دسترسی به کانال مطابق شکل شماره ۳-۲۶، از یک عدد مالتی‌پلکسر 74HC4052 استفاده شد.[۳۵]

وظیفه‌ی این مالتی پلکسـر مدیریت دسـترسـی به کانال توسـط میکروکنتلر مرکزی اسـت. کادر خاکستری رنگ بیت‌های آدرس‌دهی دودویی این مالتی‌پلکسـر را در مدار نشـان می‌دهد. هر زمان میکروکنتلر مرکزی بخواهد با یک بخش ارتباط داشـته باشـد (ارسـال و دریافت اطلاعات)، کافی است آدرس آنرا بر روی این دو بیت اعمال نماید، در آن صورت یک اتصال کوتاه میان RXD میکروکنتلر مرکزی (کادر قرمز) و TXD میکروکنتلر بخش مورد نظر(کـادر نـارنجی) ایجاد می‌شـــود. در اثر این اتصـــال کوتاه، میکروکنتلر مرکزی می تواند از بخش مورد نظر اطلاعات دریافت نماید. همچنین یک اتصال کوتاه میان TXD میکروکنتلر مرکزی (کادر قرمز) و RXD میکروکنتلر بخش مورد نظر(کادر سبز) ایجاد می‌شود. در اثر این اتصال کوتاه، میکروکنتلر مرکزی می‌تواند به بخش مورد نظر اطلاعات ارسال نماید. با وجود اینکه این دو اتصال کوتاه هم‌زمان با یکدیگر ایجاد می‌شـوند، اما به دلیل آنکه پیکربندی ارتباط سـریال در این پروژه به صـورت نیمه دوطرفه (Half-Duplex) می‌باشـد، ارسـال و دریافت اطلاعات به صـورت هم‌زمان امکان پذیر نیسـت.[۳۶]کادر بنفش رنگ مدار تغذیه این بخش را نشان می‌دهد. این مدار تغذیه اختلاف پتانسیل ۱۲ ولت باتری الکترونیک را به ۵ ولت در خروجی خود تبدیل می‌کند.




*شکل ۳- ۲۶ شماتیک مدار مالتی‌پلکسر*

دو میکروکنتلر که خواهان برقراری ارتباط سریال با یکدیگر هستند، باید نرخ ارسال بیت (Baud Rate) و فرکانس پالس ساعت آنها با یکدیگر برابر باشند. گرچه در صورتی که فرکانس‌ها با یکدیگر برابر نباشند، دو میکروکنتلر باز هم می‌توانند با یکدیگر از طریق ارتباط سریال مذاکره کنند، در این حالت به صورت تجربی ثابت شد که باید حاصلضرب نرخ ارسال بیت و فرکانس پالس ساعت یک میکرو، با دیگری برابر باشد. برای مثال دو میکروکنتلر که اولی دارای فرکانس ۱ مگاهرتز و نرخ ارسال بیت ۴۸۰۰ بیت در ثانیه است می‌تواند با یک میکروکنتلر که دارای فرکانس ۸ مگاهرتز و نرخ ارسال داده ۶۰۰ بیت در ثانیه است ارتباط سریال برقرار کند.

برای جلوگیری از ایجاد خطا در ارتباط سریال به دلیل عدم رعایت قوانین پاراگراف قبل و همچنین استانداردسازی پروتکل‌ها و ارتباطات میان پردازنده ها، **فرکانس تمامی میکروکنتلرها در این پروژه ۸ مگاهرتز و نرخ تبادل بیت تمامی آنها ۹۶۰۰ بیت در ثانیه تنظیم شد.**

### ۳-۴-۸ راه‌اندازی میکروکنتلر اصلی DSPR

میکروکنتلر اصلی روبات مرز پردازشی میان روبات و دنیای خارج از آن است. همانطور که در بخش های قبل گفته شد، DSPR از شیوه پردازش اطلاعات توسط یک عامل در خارج از روبات استفاده می کند. میکروکنتلر اصلی وظیفه‌ی دریافت اطلاعات از عامل، اسال اطلاعات به عامل و تعامل با سایر میکروکنتلرهای DSPR را دارد.

شیوه‌ی ارتباط عامل با DSPR بر مبنای مدل درخواست-پاسخ می‌باشد.[۳۷] بر اساس این مدل، عامل درخواست یا فرمان خود را برای روبات می‌فرستد و روبات به درخواست پاسخ می‌دهد یا به فرمان عمل می‌کند. این درخواست می‌تواند جهت پردازش اطلاعات صورت گیرد. برای نمونه، عامل از روبات می‌خواهد تا عرض مختصات خود را برای عامل ارسال نماید. میکروکنتلر مرکزی که کانال ارتباط سریال خود را بر روی فرستنده-گیرنده بیسیم تنظیم کرده بود، این درخواست را می‌شنود. سپس میکروکنتلر مرکزی با بخش انکودر ارتباط برقرار می‌کند (کانال را از اختیار فرستنده-گیرنده بی‌سیم خارج می‌کند و به بخش انکودر اختصاص می‌دهد) و عرض مختصات روبات را دریافت می‌کند. پس از دریافت عرض مختصات روبات، میکروکنتلر مرکزی دوباره کانال را از انکودر می‌گیرد و به فرستنده-گیرنده بی‌سیم می‌دهد. سپس عرض مختصات روبات را برای آن می‌فرستد. پس از آنکه این عملیات درخواست و پاسخ انجام شد، میکروکنتلر مرکزی دوباره از طریق فرستنده-گیرنده بی‌سیم به خط گوش می‌دهد و منتظر دریافت فرمان یا درخواست جدید از عامل می‌گردد.

در مورد فرمان نیز روال کاملا مانند درخواست است، با این تفاوت که در مورد انجام بعضی از فرمان‌ها، میکروکنتلر مرکزی پاسخی به عامل ارسال نمی‌کند، بلکه فقط فرمان را انجام می‌دهد و دوباره برای دریافت فرمان جدید به خط گوش می‌دهد.

شکل ۳-۲۷، تصویر بخشی از برنامه‌ی تابع main میکروکنتلر مرکزی را نشان می‌دهد. این تصویر گواهی بر توضیحات این بخش است. این میکروکنتلر در یک حلقه‌ی بی‌نهایت while(1) (خط ۱۵۷) مرتبا به کانال گوش می‌دهد. به محض دریافت یک دستور یا درخواست به صورت کاراکتر (خط ۱۶۱) ، شروع به رمزگشایی آن با استفاده از دستور switch-case می‌کند. شکل ۳-۲۸، لیست کاراکترهای کنترلی معتبری را که عامل می‌تواند به عنوان درخواست یا دستور برای DSPR ارسال نماید نشان می‌دهد.

به عنوان نمونه اگر کاراکتر دریافتی 'J' باشد، مطابق خط ۱۶۴ شکل ۳-۲۷، تابع up() برای حرکت رو به جلوی روبات فراخوانی می‌شود. در خط ۱۶۵ یک دیود نورانی برای نشان دادن وضعیت روبات (حرکت رو به بالا) روشن می‌شود. سپس در خط ۱۶۷ میکروکنتلر از دستور switch-case خارج می‌شود و به خط ۱۶۱ باز می‌گردد و شروع به گوش دادن خط برای دریافت دستور جدید می‌کند.

```
while (1)
{
  // Place your code here
  PORTC.1 = 1;
  f = getchar();
  switch(f)
  {
   case 'J':
   up();
   delay_ms(1000);
   //stop();
   break;

   case 'D':
   down();
   delay_ms(1000);
   //stop();
   break;


   case 'R':
   right();
   delay_ms(1000);
   //stop();
   break;

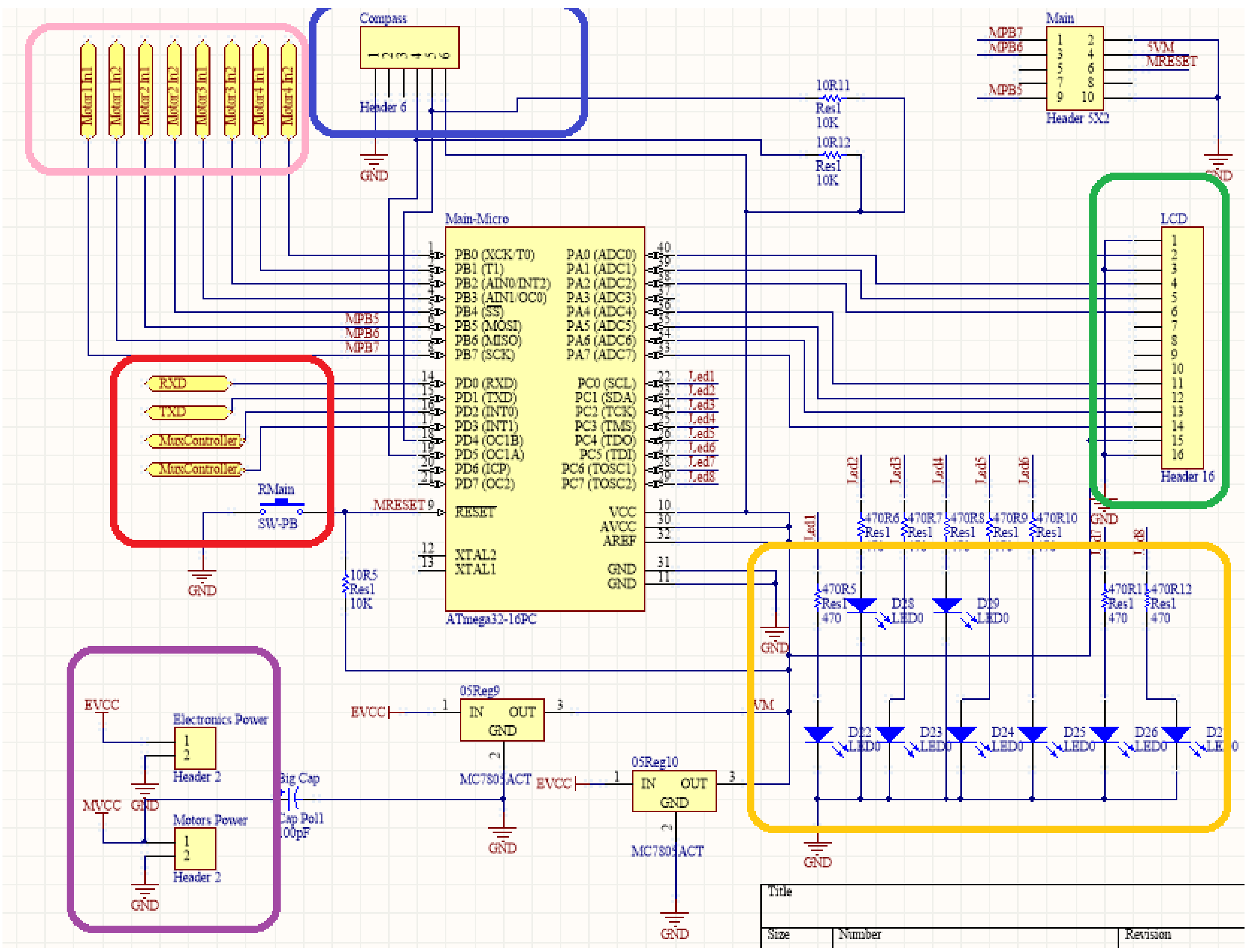
   case 'L':
   left();
   delay_ms(1000);
   //stop();
   break;
```

*شکل ۳- ۲۷بخشی از تابع main میکروکنترلر مرکزی، در این بخش از تابع، میکروکنترلر مرکزی مرتبا به خط گوش می‌دهد و منتظر دریافت یک کاراکترکنترلی می‌شود. پس از دریافت کاراکتر کنترلی آنرا رمزگشایی کرده و به انجام وظیفه می‌پردازد.*

| نام کاراکتر | دستور / درخواست | مفهوم |
|---|---|---|
| J | دستور | حرکت روبات به سمت بالا |
| D | دستور | حرکت روبات به سمت پایین |
| R | دستور | حرکت روبات به سمت راست |
| L | دستور | حرکت روبات به سمت چپ |
| S | دستور | حرکت سرو موتور جلو روبات به اندازه‌ی یک گام از پیش تعریف شده |
| s | دستور | حرکت سرو موتور عقب روبات به اندازه‌ی یک گام از پیش تعریف شده |
| U | درخواست | مقدار فاصله ی سنجیده شده توسط حسگر فراصوت جلو |
| u | درخواست | مقدار فاصله ی سنجیده شده توسط حسگر فراصوت جلو |
| X | درخواست | طول مختصات کنونی روبات |
| Y | درخواست | عرض مختصات کنونی روبات |
| Q | دستور | حرکت سرو موتورها به زاویه صفر درجه |
| Z | دستور | توقف کامل روبات |

*شکل ۳- ۲۸ لیست کاراکترهای کنترلی معتبر برای DSPR*

شـکل ۳-۲۹ تصـویر شـماتیک مدار میکروکنتلر اصـلی DSPR را نشـان می‌دهد. بخش بنفش رنگ محل اتصـال باتری‌های بخش الکترونیک و باتری بخش موتورها را به مدار نشـان می‌دهد. کادر صـورتی رنگ ارتباط ورودی/خروجی میکروکنتلر اصـلی را به درایور موتورها نشـان می‌دهد. محل اتصـال یک LCD کاراکتری ۱۶*۲ به مدار از قبل پیش‌بینی شده است تا در صورت لزوم برای مانیتورینگ از آن استفاده شـود. کادر زرد رنگ دیودهای نورانی را نمایش می‌دهند. این دیودها با هدف نشـان‌دادن وضـعیت DSPR اسـتفاده می‌شوند و وضعیت‌هایی از قبیل چرخش سـروها، حرکت روبات در جهات مختلف، کارکرد درسـت انکودرها و انحراف روبات از راسـتای اصـلی را نشانی می‌دهند. همانند تمامی شـماتیک‌هایی که تا به حال بررسی شدند، کادر قرمز درگاه‌های ارتباطی میکروکنتلر مرکزی را با سایر بخش ها از طریق ارتباط سریال بیان می‌کند. عالوه بر خطوط RXD و TXD که مربوط به ارتباط سریال هستند، میکرو کنتلر مرکزی دارای ۲ خط کنترلی دیگر می‌باشد که در کادر قرمز مشـخص هسـت. این خطوط برای مدیریت دسـترسـی به کانال ارتباط سـریال استفاده می‌شوند و در بخش ۳-۴-۷ به صورت کامل توضیح داده شد.




شکل ۳- ۲۹ شماتیک میکروکنتلر مرکزی روبات

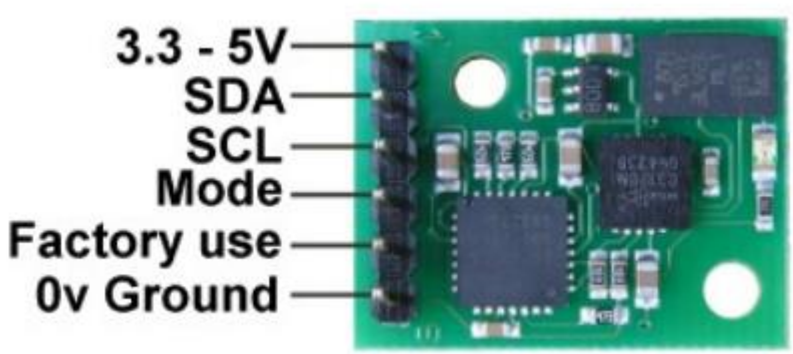
### ۳-۴-۹راه اندازی قطب نمای DSPR

در شکل ۳-۲۹، کادر آبی رنگ مشخص کننده‌ی محل اتصال قطب نما به روبات می‌باشد. این قطب نما با هدف تشخیص انحراف روبات از راستای اصلی حرکت خود صورت تعبیه شده است. انتخاب قرار دادن قطب‌نما به عنوان بخش تشخیص دهنده‌ی سیستم اصلاح حرکت برای DSPR، پس از مطالعه بر روی روش‌های رایج اصلاح حرکت در روبات‌های هوشمند مشابه انجام شد[۱۲, ۱۶, ۱۷, ۳۸]. انحراف از راستای اصلی حرکت می تواند در اثر برابر نبودن سرعت یا گشتاور چرخ ها، حرکت در سطح شیب دار، یکسان نبودن شیب زمین در محل هر یک از چرخ‌ها، وجود دست‌انداز در سر راه روبات و مواردی مانند این‌ها باشد. شیوه‌ی اصلاح حرکت روبات به این صورت است که در هنگام شروع اولیه‌ی حرکت، زاویه‌ی اولیه‌ی خود را با جهت مثبت محور y ها صفر در نظر می‌گیرد. سپس شروع به حرکت کرده و مرتبا زاویه‌ی جدید خود را با محور y ها اندازه‌گیری می‌کند. هرگاه مقدار این زاویه از مقدار قابل تنظیم و از پیش تعریف شده‌ای که در اصطلاح به آن زاویه آستانه انحراف گفته می‌شود بیشتر شد، روبات شروع به اصلاح حرکت می‌نماید.

الگوریتم کلی اصلاح حرکت DSPR به این صورت است که درجهت مخالف انحراف، شروع به چرخیدن می‌نماید. این کار تا زمانی ادامه پیدا می‌کند که قطب نما زاویه‌ای کمتر از زاویه‌ی آستانه‌ی انحراف گزارش نماید. اصلاح حرکت DSPR فقط در هنگام حرکت در راستای محور y ها (حرکت به بالا و پایین) انجام می‌شود. زیرا فرض بر این است که حرکت در راستای محورx ها (حرکت به چپ و راست) به صورت جزئی و با مقادیر مسافت کم انجام می‌پذیرد. گرچه پس از پایان حرکت در راستای محور xها و شروع مجدد در راستای محور yها در صورت وجود انحراف، سیستم تشخیص انحراف فعال می‌شود.

پس از مطالعه بر روی قطب‌نماهای موجود در بازار بومی، سه عدد از ماژول های قطب‌نما تحت معیارهای راه‌اندازی ساده، پروتکل استاندارد، قیمت مناسب و از همه مهمتر دقت بالا انتخاب شدند. این مازول‌های قطب نما عبارتند از CMPS 03، CMPS 09 و ZCT245AN.[۳۹] پس از انجام آزمایش بر روی این ماژول‌ها، ماژول CMPS 09 به دلیل عدم حساسیت به لرزش، دقت بسیار بالا و نویزپذیری کمتر نسبت به دو ماژول دیگر، به عنوان ماژول مناسب برای سیستم اصلاح حرکت روبات انتخاب شد.

ماژول CMPS 09 از لحاظ راه اندازی شبیه ماژول SRF08 می‌باشد. کاربرد اصلی این ماژول به صورت قطب نما و یا شتاب سنج سه محوره می‌باشد. شکل ۳-۳۰ نمایی از این ماژول را به همراه راهنمای پایه‌های آن نشان می دهد.




*شکل ۳- ۳۰ قطب نمای .CMP 09 به همراه نقشه‌ی پایه‌های آن*

راه‌اندازی اولیه‌ی این ماژول دقیقا همانند آنچه در مورد راه‌اندازی ماژول SRF08 در بخش ۳-۴-۲ گفته شد، می‌باشد. پایه‌های تغذیه باید به پایه‌های نظیر در شماتیک شکل ۳-۳۰ متصل شوند. پایه‌های SDA و SCL نیز باید به پایه‌های تعریف شده در پیکربندی برنامه متصل شوند. شکل ۳-۳۱ جدول راهنمای دستورات کنترلی این ماژول را نشان می‌دهد. در این جدول نوع، کار و شماره‌ی هر یک از ثبات‌های داخل ماژول آورده شده است. به عنوان مثال ثبات شماره ۱ شامل اطلاعات قطب‌نما به صورت یک عدد بین ۰ تا ۲۵۵ حول ۳۶۰ درجه کامل اطراف است.

| Register | Function |
|---|---|
| 0 | Software version |
| 1 | Compass Bearing as a byte, i.e. 0-255 for a full circle |
| 2,3 | Compass Bearing as a word, i.e. 0-3599 for a full circle, representing 0-359.9 degrees. |
| 4 | Pitch angle - signed byte giving angle in degrees from the horizontal plane |
| 5 | Roll angle - signed byte giving angle in degrees from the horizontal plane |
| 6 | Unused |
| 7 | Unused |
| 8 | Unused |
| 9 | Unused |
| 10,11 | Magnetometer X axis raw output, 16 bit signed integer with register 10 being the upper 8 bits |
| 12,13 | Magnetometer Y axis raw output, 16 bit signed integer with register 12 being the upper 8 bits |
| 14,15 | Magnetometer Z axis raw output, 16 bit signed integer with register 14 being the upper 8 bits |
| 16,17 | Accelerometer X axis raw output, 16 bit signed integer with register 16 being the upper 8 bits |
| 18,19 | Accelerometer Y axis raw output, 16 bit signed integer with register 18 being the upper 8 bits |
| 20,21 | Accelerometer Z axis raw output, 16 bit signed integer with register 20 being the upper 8 bits |
| 22 | Command register |

*شکل ۳- ۳۱ جدول دستورات کنترلی ماژول CMPS 09*

شکل ۳-۳۲ یک نمونه برنامه برای ارتباط با ماژول CMPS 09 را نشان می‌دهد. این برنامه برای خواندن محتوی ثبات شماره ۱ مورد استفاده قرار می‌گیرد. پس از پایان اجرای برنامه، محتوی ثبات شماره ۱ که همان زاویه اندازه‌گیری شده می‌باشد. در داخل متغییر amount قرار می‌گیرد.

```
while (1)
        {

        i2c_start();
        i2c_write(0xc0);
        i2c_write(0x01);
        i2c_start();
        i2c_write(0xc1);

        amount=i2c_read(0);

        i2c_stop();

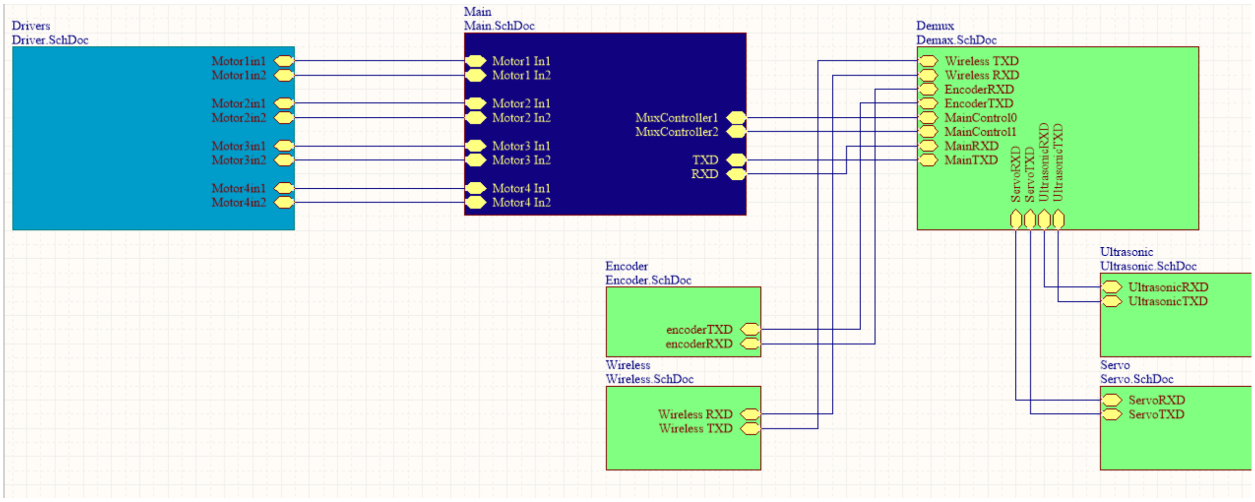
```

*شکل ۳- ۳۲ نمونه ای از ارتباط با ماژول CMPS 09*

## ۳-۵ طراحی مدار ماژولار

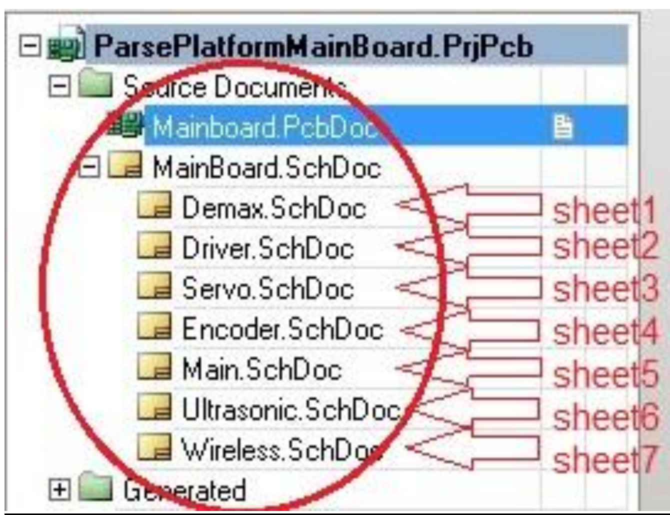
در بخش ۲-۶-۱۱ ذکر شد که طراحی مدار روبات به صورت ماژولار است. در بخش ۴-۳ مشاهده شد که مدار تمامی بخش‌های روبات به صورت کاملا مستقل از یکدیگر هستند. این بخش‌ها حتی در سیستم تغذیه‌ی خود نیز مستقل بودند. هر بخش از طریق ارتباط سریال با میکروکنتلر مرکزی در ارتباط بود. در واقع این ارتباط سریال درگاه ارتباطی هر بخش با بیرون از خود بود. مدار هر بخش به صورت یک جعبه‌سیاه که فقط یک درگاه ورودی/خروجی ارتباط سریال دارد، قابل تصور است. دلیل این موضوع تلاش برای پیاده‌سازی مدار روبات به صورت کاملا ماژولار بود. با وجود اینکه هر بخش شماتیک مدار کاملا مجزا و مستقل داشت و به صورت یک ماژول در نظر گرفته می‌شد، اما هنوز هم مفهوم ماژولار به صورت کامل پیاده‌سازی نشده بود. زیرا روابط میان ماژول‌ها به طور واضح مشخص نشده بود.

برای پیاده‌سازی طراحی مدار ماژولار از فناوری طراحی چند صفحه (Multi-Sheet Design) نرم افزار Altium Designer 10 استفاده شد.[۴۰] این فناوری اجازه می‌دهد تا شماتیک‌های مختلف مدار که طبق استانداردهای طراحی چند صفحه آماده شده‌اند، به عنوان واحدهایی مستقل و به صورت جعبه‌های سیاه در کنار هم قرار بگیرند. از طریق سیم‌کشی میان این جعبه‌های سیاه می‌توان روابط میان آنها را مشخص کرد. شکل شماره ۳-۳۲ پیاده‌سازی طراحی مدار روبات را به صورت ماژولار، با استفاده از فناوری طراحی چند صفحه‌ای نشان می‌دهد. در این شکل بخش‌های مختلف مدار که شماتیک‌های آنها در بخش۳-۴ مورد بررسی قرار گرفتند، به صورت جعبه‌هایی رنگی و بدون پرداختن به جزئیات پیاده‌سازی داخل هریک از آنها در کنار یکدیگر قرار گرفته‌اند. همانطور که در شکل ۳-۳۳ قابل مشاهده است، هر جعبه تنها با ۳ ویژگی مشخص شده است: نام، نام فایل شماتیک مربوط و نقاط زرد رنگی در داخل جعبه که معرف درگاه های ورودی/ خروجی آن می‌باشند. مهمترین نقش طراحی چند صفحه که در این شکل نیز قابل مشاهده است، تعیین روابط متناظر و متقابل میان این جعبه‌هاست.



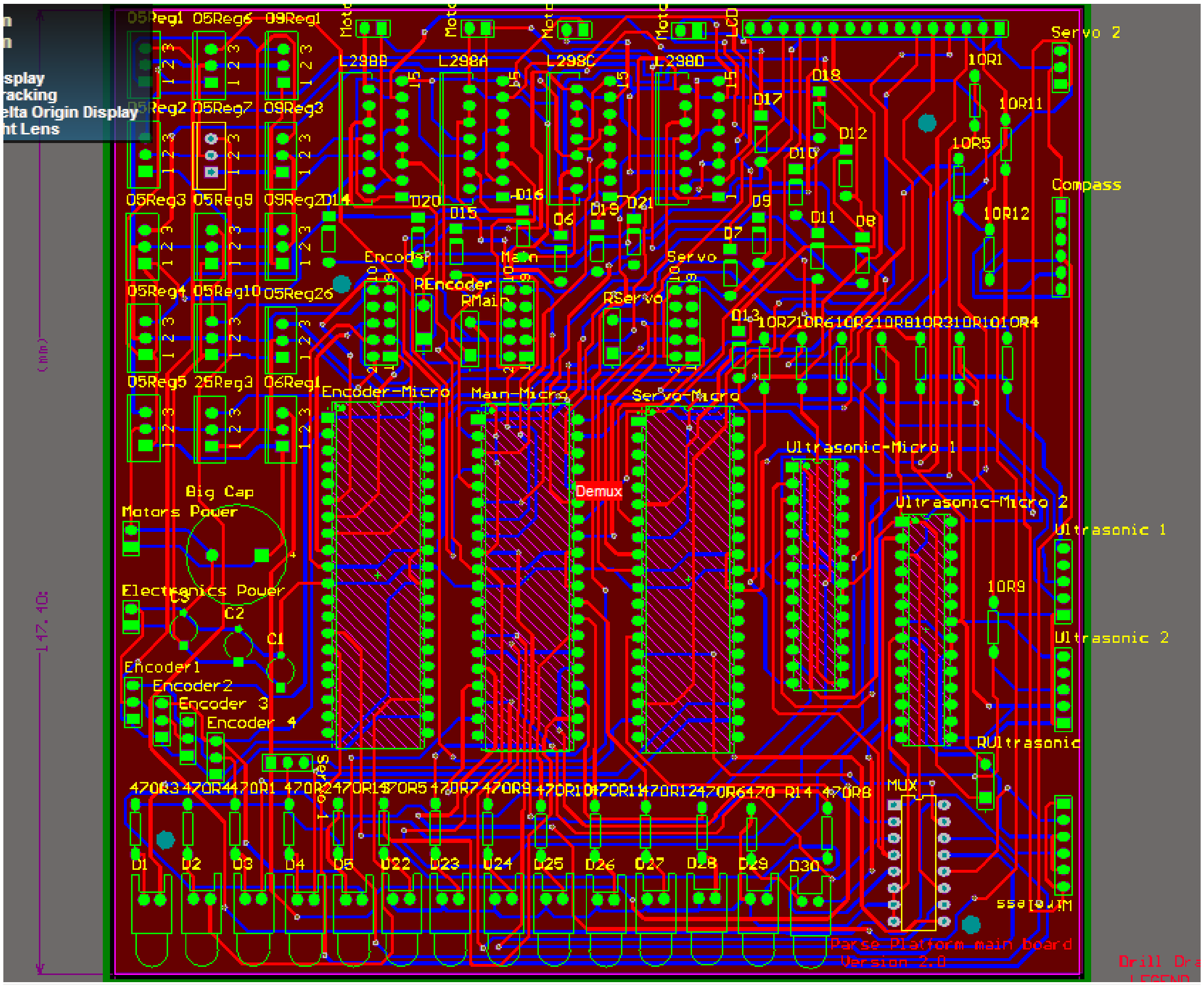

*شکل ۳- ۳۳ طراحی مدار روبات با استفاده از فناوری طراحی چند صفحه، هر بخش به صورت جعبه‌های رنگی مشخص شده است* و از طریق درگاههای زد رنگی که درون هر بخش قرار دارد، با یکدیگر به ارتباط می‌پردازند.

شـکل ۳-۳۳ در واقع در ذات و ماهیت یک شماتیک است و سایر شماتیک‌ها به صورت زیر مجموعه‌های آن تعریف می‌شوند. شکل ۳-۳۴ نشـان‌دهنده‌ی این ادعا می‌باشد. در فناوری طراحی چند صفحه‌ای، هر یک از شماتیک‌ها که زیرمجموعه‌ی شماتیک اصلی ( در اینجا شکل ۳-۳۳) می‌شوند، به عنوان یک صفحه کاملا مستقل در نظر گرفته می‌شوند.

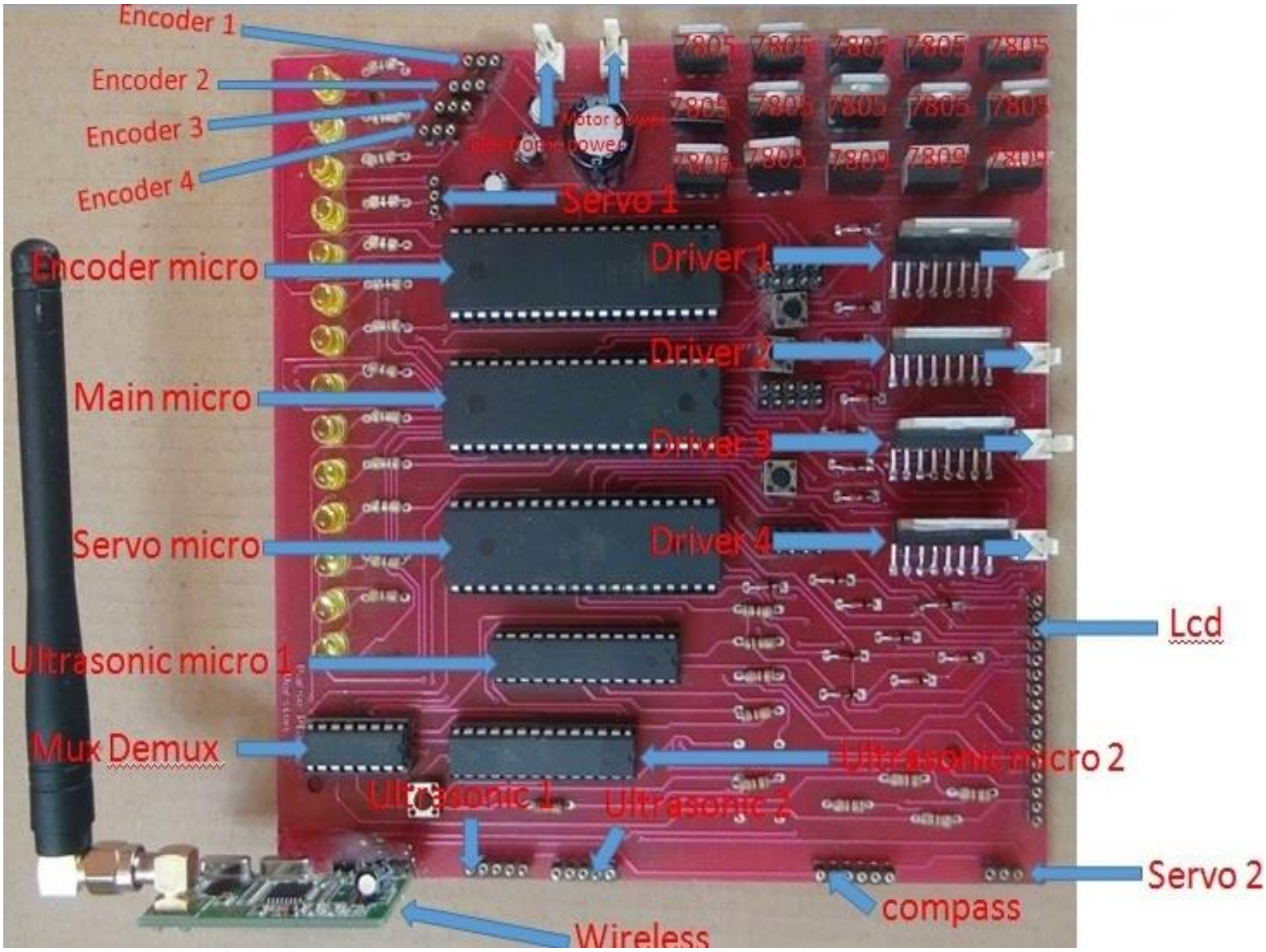



*شکل ۳- ۳۴ معرفی شماتیک‌ها به عنوان صفحه های فناوری طراحی چند صفحه‌ای، در اینجا همه‌ی شماتیک‌ها زیرمجموعه‌ی شماتیک MainBoard می‌باشند.*

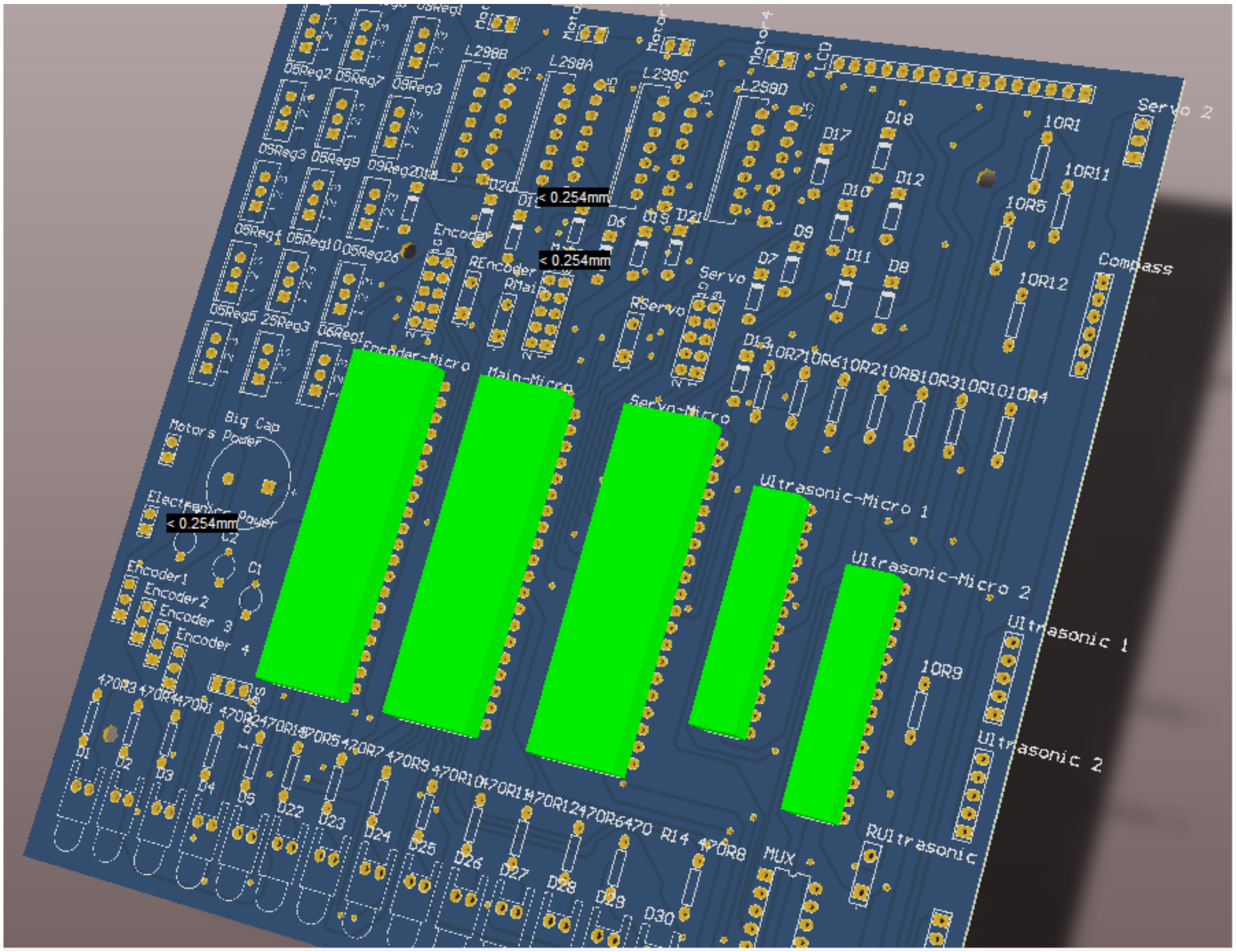
شکل‌های ۳-۳۵ و ۳-۳۶ پیاده‌سازی‌های **طراحی نمونه‌ی نهایی** PCB و برد مدار چاپی فیزیکی DSPR را که با فناوری طراحی چند صـفحه پیاده‌ســازی شـدند، نشان می‌دهند. همچنین شکل ۳-۳۷ نمونه ی سه‌بعدی PCB مدار اصلی DSPR را نشان می‌دهد. این شکل برای کمک به تجسم PCB اصلی و به عنوان راهنمایی برای درک بهتر شکل ۳-۳۵ آمده است.[۴۱]




شکل ۳- ۳۵ نمونه ی نهایی طراحی شده PCB روبات DSPR




*شکل ۳- ۳۶ پیاده سازی بورد اصلی روبات DSPR با استفاده از فناوری طراحی چند صفحه ( این بورد نمونه ی پیاده سازی شده ی شکل های ۳-۳۳و ۳-۳۴ و ۳-۳۵ می‌باشد)*




*شکل ۳- ۳۷ نمونه‌ی سه‌بعدی PCB روبات*

## ۳-۶ برنامه نمایشی(Demo)

هر دســتگاهی که قادر به پشــتیبانی از ارتباط ســریال باشــد می‌تواند در فرکانس ۸ مگاهرتز و نرخ تبادل بیت ۹۶۰۰ بیت در ثانیه، با DSPR مطابق با کاراکترهای کنترلی اســتانداردی که در شــکل شــماره ۳-۲۸ ذکر شدند، تبادل اطلاعات نماید. این ارتباط مستقل از هرگونه زبان برنامه‌نویســی می‌باشد. DSPR می‌تواند با هر زبان برنامه‌نویسی و برنامه‌ای که خروجی را به صورت فرمانی مطابق با استاندارد شکل ۳-۲۸ تولید نماید و قادر به پذیرفتن پاسخ از DSPR مطابق با همان استاندارد باشد، ارتباط برقرار کند.

به عنوان یک نمونه، یک برنامه نمایشــی برای DSPR با زبان #C نوشــته شــد. کد برنامه‌ی این نسـخه نمایشــی در فایل‌های پیوست موجود می‌باشـد. این برنامه نمایشــی به صـورت یک کلاس پیاده‌سازی شد و دارای توابعی برای برقراری ارتباط سریال با DSPR می‌باشد. این توابع شامل خواندن و نوشتن بر پورت سریال DSPR، تابع دست‌دادن (Handshaking)، تنظیم Baud Rate، تنظیم بیت پایان و بیت‌های Parity می‌باشـند. همچنین یک تابع برای پویش ۱۸۰ درجه‌ی بالا یا پایین DSPR توسط حسگرهای فراصوت نوشته شد (ScanSweep). با استفاده از این توابع الگوریتم Sweep برای جاروب کردن یک محیط فرضی برای DSPR پیاده‌سازی شده است.

این Demo یک نسـخه‌ی آزمایشـی می باشـد و البته دارای نقایص جزئی می‌باشـد. هدف از طراحی و پیاده سـازی این Demo صـرفا آزمـایش توانـایی‌هـای DSPR برای تبـادل اطلاعـات و برنـامـه ریزی از طریق یـک کـامپیوتر شـــخصـــی اســـت. چون هـدف این پروژه بر اســتانداردســازی‌های DSPR تاکید بیشــتری داد، لذا از صرف زمان بیشتر برای نوشتن نسخه های Demo فعلا صرف نظر شد. این موضوع جزء برنامه‌های آینده این پروژه‌ی می باشد.

## ۳-۷ جمع‌بندی فصل سوم

در این فصل به پیاده‌سازی قطعات و بخش‌های الکترونیک، مکانیکی و نرم‌افزاری DSPR پرداخته شد و جزئیات پیاده‌سازی‌ها تا حد امکان به صورت کاملا شفاف بیان گردید. همچنین ذکر شد که برای آزمایش توانایی‌های DSPR یک نسخه‌ی نمایشی برای آن نوشته شده است که با نام Demo در فایل‌های پیوست موجود است.

# فصل چهارم

# نتایج آزمایشات انجام شده بر روی سکو

## ۴-۱ معرفی دو مفهوم کلیدی در ارزیابی کارایی DSPR

در این بخش به معرفی دو واژه‌ی کلیدی که در سنجش کارایی DSPR استفاده می‌شوند، پرداخته می شود:

### ۴-۱-۱ وضوح (Resolution)

در DSPR تعداد نقاطی را که هر حسگر فراصوت در یک چرخش ۱۸ کامل سرو موتور نظیر خود، نمونه‌برداری می‌کند، وضوح نامیده می‌شود.

### ۴-۱-۲ گام (Step)

فاصله‌ی میان هر دو نقطه‌ی متوالی را که DSPR در هنگام حرکت و جابجایی فیزیکی در آن نقاط می‌ایستد و حسگرهای فراصوت آن شروع به پویش ۳۶۰ درجه یا ۱۸۰ درجه اطراف خود می‌کنند، گام نامیده می‌شود.

## ۴-۲ سنجش کارایی DSPR

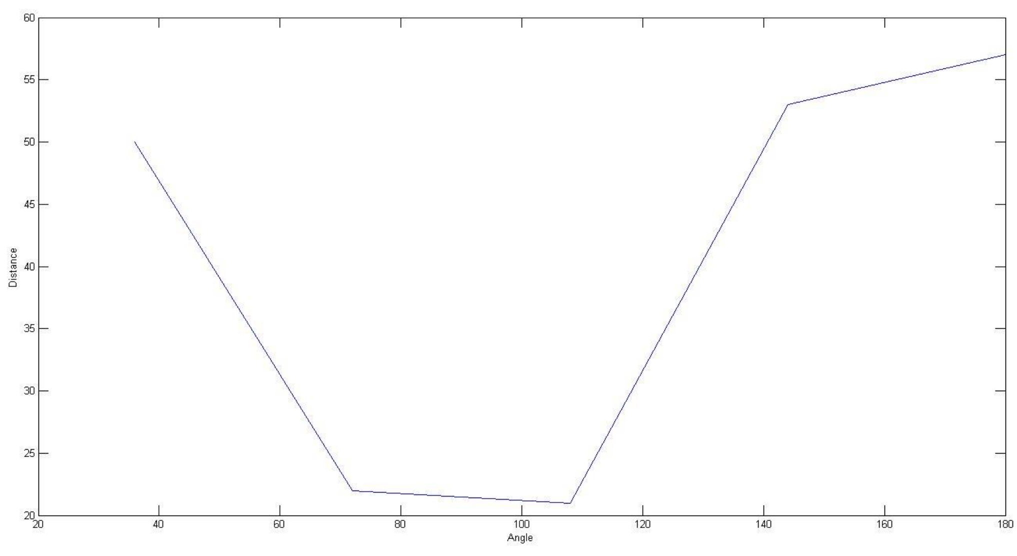
به طور خاص، کارایی DSPR با توجه به تصویری که از محیط اطراف می‌سازد، سنجیده می‌شود. تاکنون معیار عددی خاصی برای محاسبه‌ی رقمی کارایی DSPR در نظر گرفته نشده است. زیرا DSPR یک سکو است. روباتی با چنین ویژگی نباید به صورت کلی سنجیده شود. چنین کاری مفهوم سکو بودن DSPR را کمرنگ می‌کند. سنجش کارایی DSPR از طریق الگوریتم‌های خاصی که به جای روبات‌های خاص اجرا می‌کند و در مقایسه با آن روبات‌ها سنجیده می‌شود. با این حال، با توجه به اینکه DSPR با هدف حل مشکل جمع‌آوری اطلاعات در خانواده‌ای از روبات‌های هوشمند طراحی شده است، می توان بر روی کیفیت تجسمی که از محیط خلق می‌کند، به عنوان معیاری از کارایی آن بحث کرد.

### ۴-۲-۱ چند مثال از پویش محیط با وضوح مشخص

شکل ۴-۱ تصویری از DSPR را در حال انجام پویش در مقابل یک سطح (درب یخچال) نشان می‌دهد. شکل ۴-۲، تجسمی را که DSPR با وضوح ۵ نقطه از این پویش در دو بعد خلق می‌کند، به نمایش می‌کشد. در این تصویر، DSPR به صورت موازی با درب یخچال و در فاصله‌ی ۲۲ سانتی‌متری از آن قرار دارد. در شکل ۴-۲ مشاهده می‌شود که DSPR محیط روبروی خود را تقریبا به صورت قابل درک در دو بعد مدل‌سازی کرده است. در این شکل محور xها، زاویه‌ی حسگر فراصوت از راستای قائم و محور yها فاصله‌ای را که حسگر فراصوت اندازه‌گیری کرده است، نشان می‌دهند. در مدل‌سازی شکل ۴-۲، DSPR در نقطه‌ی تقریبی در اواسط محور xها قرار دارد.



شکل ۴- ۱ پویش درب یخچال در حالتی که موازی با DSPR قرار دارد

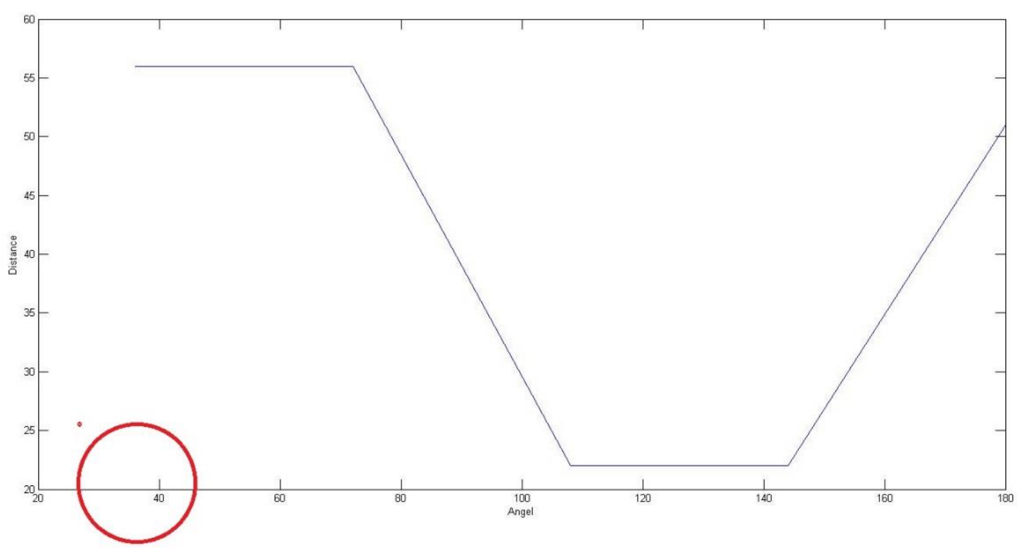



شکل ۴- ۲ تجسمی دو بعدی که DSPR پس از پویش شکل ۳-۳۸ ایجاد می کند(آنگونه که DSPR محیط را می‌بیند).

شکل ۴-۳، DSPR را در حال پویش همان درب یخچال تصویر ۴-۱ نشان می‌دهد. تفاوتی که این دو تصویر با هم دارند این است که این بار DSPR موازی درب یخچال نیست بلکه با زاویای تقریبا ۳۰ درجه با محور آن قرار دارد. فاصله‌ی حسگر فراصوت از درب یخچال ۲۴ سانتی‌متر می‌باشد. شکل ۴-۴ تجسمی را که DSPR در دو بعد خلق می‌کند، نشان می‌دهد. محل تقریبی قرارگیری DSPR در این تصویر با دایره قرمز رنگ نشان داده شده است.

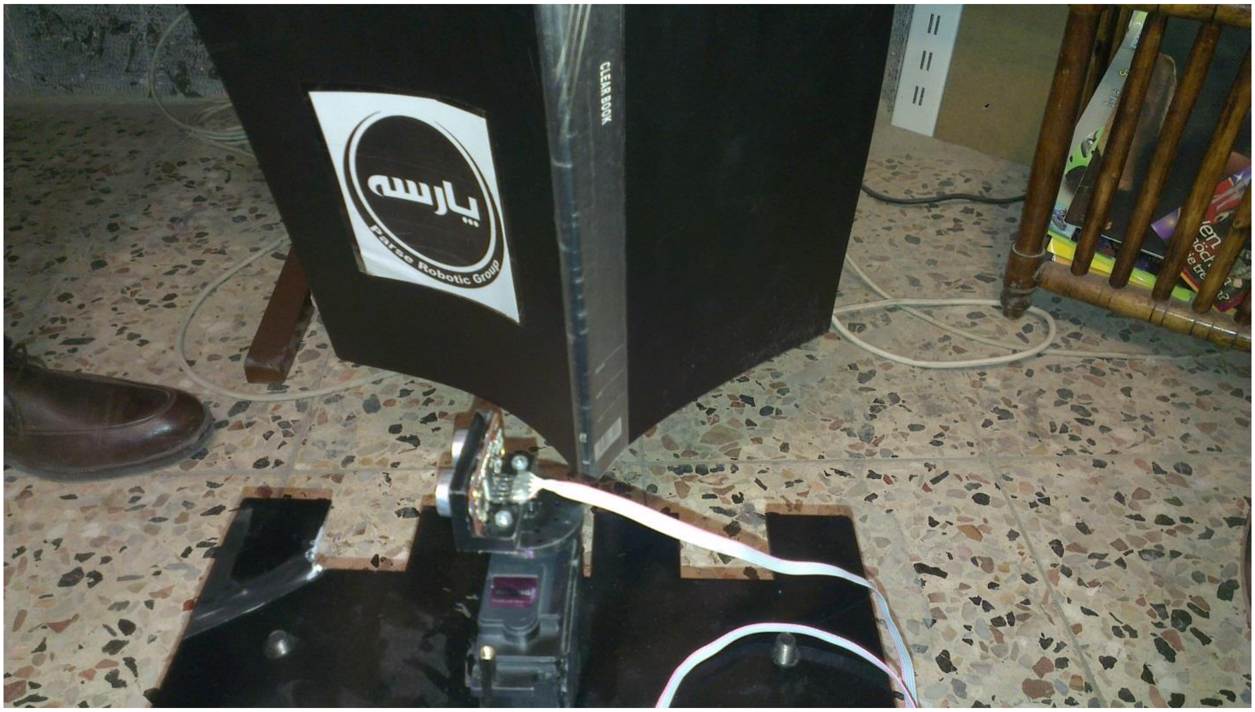


*شکل ۴- ۳ پویش درب یخچال توسط DSPR در حالتی که موازی با آن قرار ندارد*

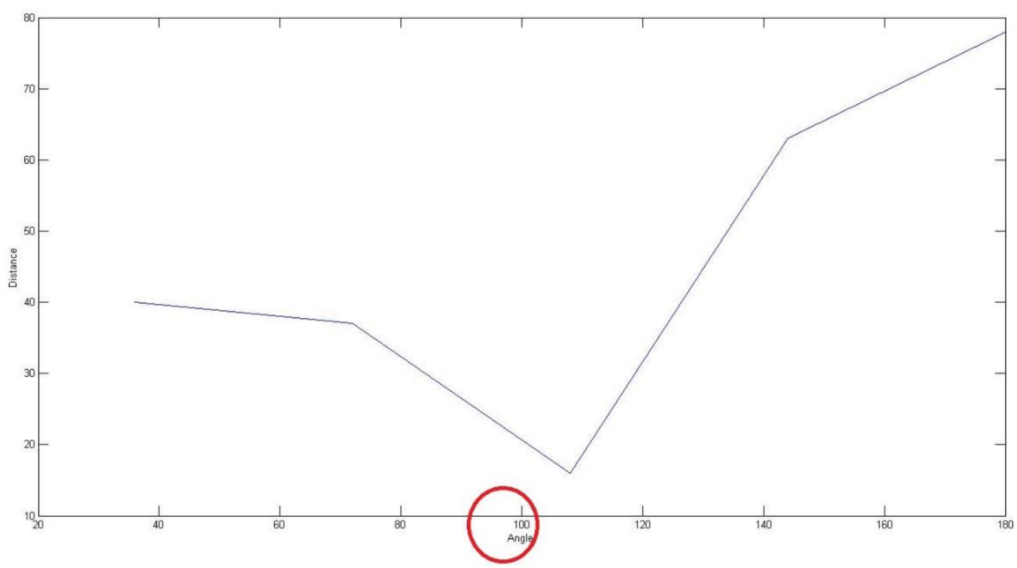



*شکل ۴- ۴ تصویری که DSPR پس از پویش شکل ۴-۳ ترسیم می کند*

در شکل ۴-۵، DSPR در حال پویش یک پوشه‌ی پلاستیکی است که به شکل خاصی(مطابق آنچه در شکل ۴-۵ مشاهده می‌شود) در مقابل DSPR قرار گرفته است. همچنین شکل ۴-۶ تصویر دوبعدی را که DSPR با وضوح ۵ نقطه از آنچه در شکل ۴-۵ می‌بینند، نشان می‌دهد. محل تقریبی قرارگیری DSPR در این تصویر با دایره قرمز رنگ نشان داده شده است.




*شکل ۴- ۵ در حال پویش جلد یک پوشه پلاستیکی و محیط اطراف*

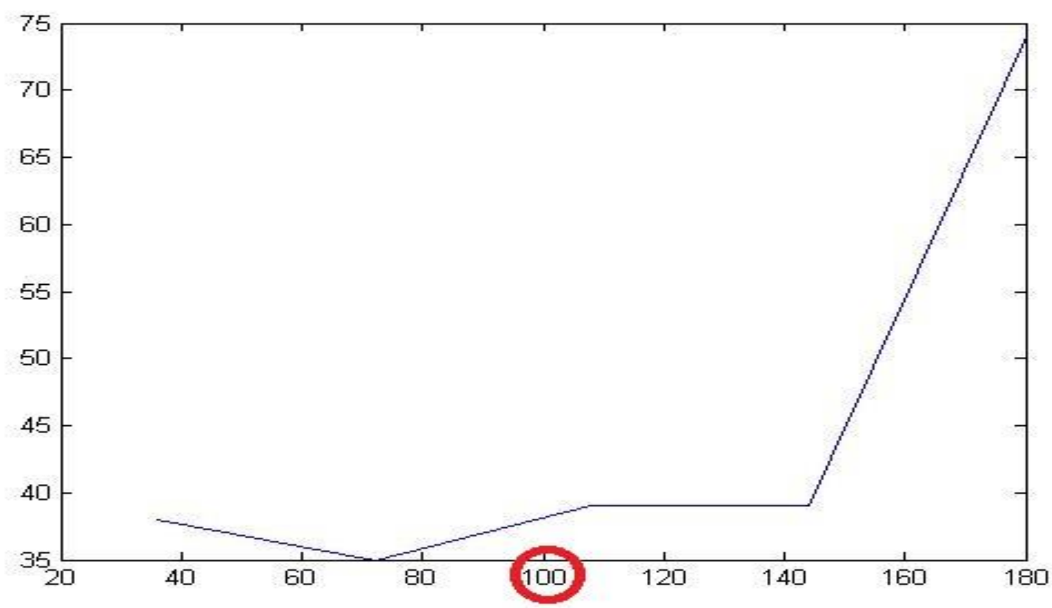



*شکل ۴- ۶ تصویری که DSPR پس از پویش شکل ۴-۵ در دوبعد ایجاد می‌کند.*

در نهایت شـکل ۴-۷، DSPR را در حال پویش یک تورفتگی نشـان می‌دهد. فاصله‌ی دقیق حسگر فراصوت تا محل تقاطح دو سطح و ایجاد تو رفتگی، ۳۸ سانتی‌متر می باشد. شکل ۴-۸ تصویر دوبعدی را که DSPR پس از پویش شکل ۴-۷ با وضوح ۵ نقطه ایجاد می‌کند، نشان می‌دهد. محل تقریبی قرارگیری DSPR در این تصویر با دایره قرمز رنگ نشان داده شده است.



*شکل ۴- ۷ DSPR در حال پویش یک سطح تورفتگی*

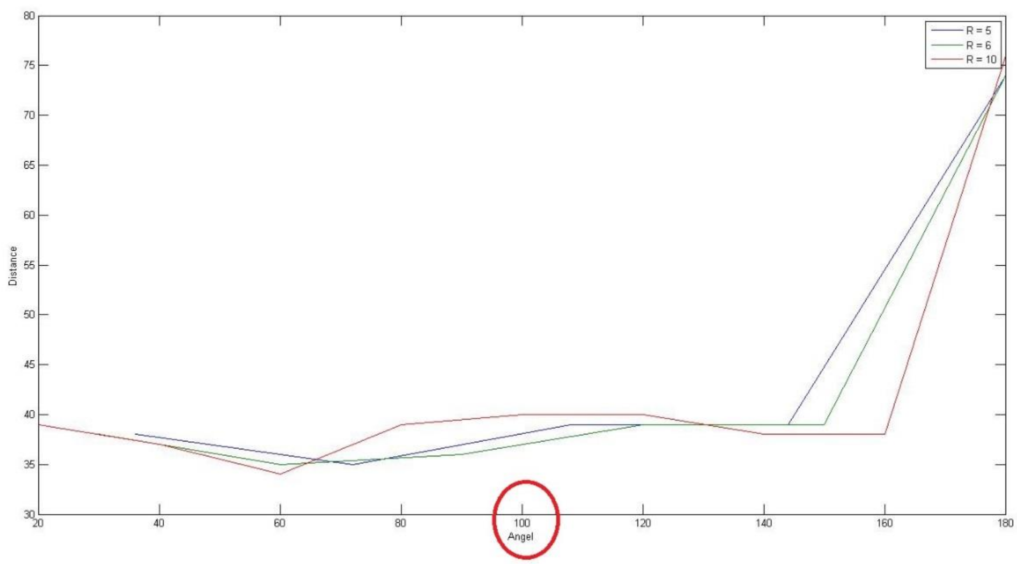



*شکل ۴- ۸ تصویر دوبعدی که DSPR از پویش شکل ۴-۷ ایجاد می‌کند*

توجه شـود که تمامی تصـاویر ۴-۱ تا ۴-۸ با فرض وضوح ۵ نقطه برای DSPR مطرح شدند. با این وجود ملاحظه می‌شود که DSPR برای خلق کلیات تصـویر به مشـکلی بر نمی‌خورد. در تمامی این پویش‌ها، هدف ترسیم تصویری با کلیات هرچه بیشتر توسط DSPR بود. زیرا فرض بر این اسـت که در مورد ماموریت‌هایی که DSPR می‌خواهد به عنوان روبات جایگزین اسـتفاده شود، کلیات بر جزئیات غلبه دارند. برای مثال در الگوریتم جاروب کردن، آنچه اهمیت دارد تشخیص مانع در بازه‌ای با خطای قابل قبول با هدف جلوگیری از برخورد با مانع است. اینکه مانع دقیقا در کجا قرار دارد، اهمیت چندانی ندارد. همچنین در الگوریتمهای نقشـه‌برداری، به طور معمول به یک بار پویش محیط از یک زاویه و یک نقطه بسـنده نمی‌شود. بلکه با چندین بار پویش محیط از نقاط مختلف و پیدایش کلیات اطلاعات و سپس خلق جزئیات با قراردادن این ویژگی‌های کلی در کنار هم، نقشـه حاصل می‌شود. به همین دلیل در این مثال‌ها، تاکیدی بر جزئیات فاصله ها و زاویه ها انجام نشد و به خلق یک تصویر و تجسم کلی از محیط اطراف پرداخته شد.

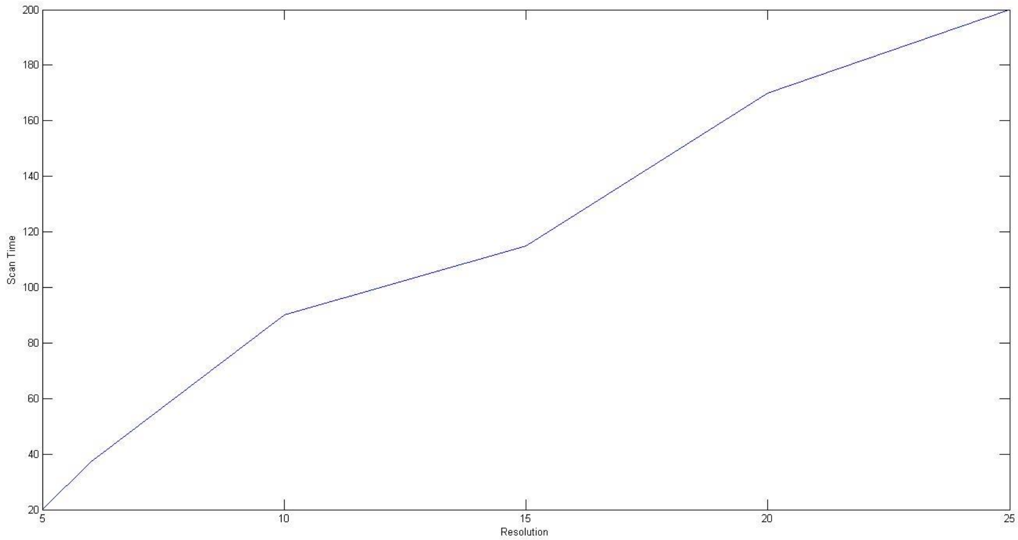
### ۴-۲-۲ بحث بر روی وضوح‌های مختلف

شـکل ۴-۹ نتایج حاصـل از پویش شکل ۴-۷ را با وضوح‌های مختلف نشان می‌دهد. نمودار آبی رنگ، وضوح ۵ نقطه، نمودار سبز رنگ وضـوح ۶ نقطه و نمودار قرمز رنگ وضـوح ۱۰ نقطه را نشـان می‌دهد. شـاید در نگاه اول این مشـاهده حاصـل شود که نمودار سبز و آبی رنگ دقیق‌تر هسـتند و در نتیجه هرچه وضـوح پایین‌تر باشـد، تجسـم از محیط واقعی‌تر اسـت. این مسئله به علت خطایی است که برای درون‌یابی نمودار توسط نرم‌افزار Matlab ایجاد می‌شود. به دلیل نبود اطلاعات در مورد بخشی‌هایی از نمودار، درون‌یابی آنها با مشکل مواجه می‌شود. این روند تا وضوح ۲۰ آزمایش شد و همچنان خطای درون‌یابی وجود داشت. به دلیل جلوگیری از شلوغی نمودار، شباهت نتایج وضوح‌های بالاتر به وضوح ۱۰ و دور شدن از موضوع اصلی وضوح‌های ۱۵ و ۲۰ در نمودار وارد نشد.




*شکل ۴- ۹ پویش تصویر ۴-۷ با وضوح های مختلف*

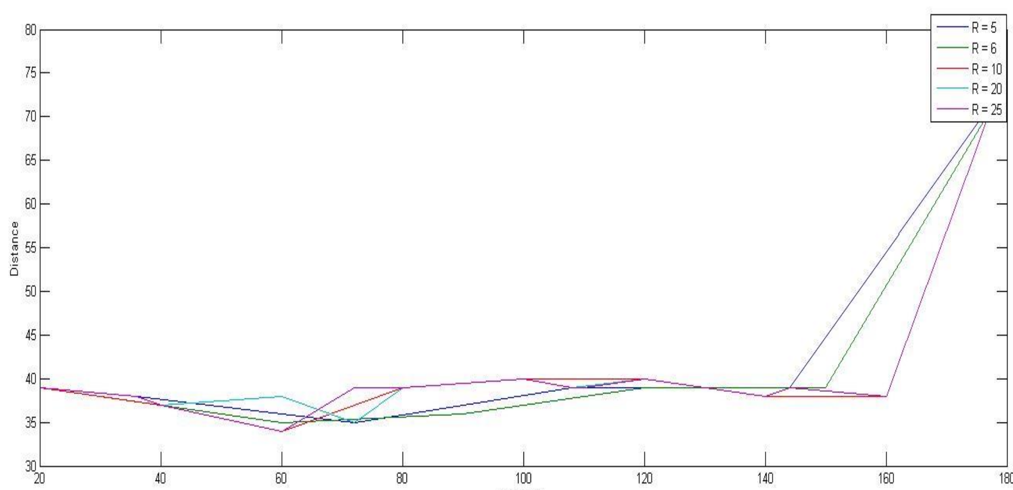
اولین نتیجه‌ای که از نمودار دریافت می‌شـود، این اسـت که هرچقدر وضـوح بیشـتر باشـد، **تصـویر دو بعدی ترسـیم شده شامل اطلاعات بیشـتری از محیط اسـت.** نتیجه‌ی دوم کمی غیر منتظره اسـت. مدت زمان پویش کامل ۱۸۰ درجه، با افزایش وضـوح به صورت تصـاعدی افزایش می‌یابد. شکل ۴-۱۰ این روند را نشان می‌دهد. محور x ها وضوح پویش و محور y ها زمان لازم برای پویش را بر حسب ثانیه نشان می‌دهد. این مسئله به خاطر تاخیرهای پردازشی DSPR در جمع آوری اطلاعات رخ می‌دهد.




شکل ۴- ۱۰ افزایش تصاعدی مدت زمان پویش با افزایش وضوح، محور افقی وضوح و محور عمودی زمان پویش را نشان می‌دهند.

در اینجا به اولین مصالحه‌ی DSPR برخورد می کنیم: با افزایش وضوح، اطلاعات بیشتری از محیط دریافت می‌شود، اما در مقابل مدت زمان یک پویش کامل ۱۸۰ درجه به صورت تصاعدی افزایش می‌یابد. به صورت تجربی نتیجه شد که  تاخیر کل عملیات **ممکن** است متناسب با مجذور وضوح زیاد شود.

حال که این مصالحه عنوان شد، می‌توان شکل ۴-۹ را کامل تر رسم کرد، زیرا تصور می‌شود تاثیری بر عدم درک مصالحه‌ی گفته شده نخواهد داشت. این تصویر، صرفا با هدف درک درست از افزایش وضوح DSPR آورده شده است. همانطور که در  تصویر ۴-۱۱ مشاهده می‌شود، افزایش وضـوح بعد از وضـوح ۱۰ عملا چیز زیادی به کلیات تصـویر اضـافه نمی‌کند. فقط باعث اصـلاح بعضی از نقاط می‌شود. به عبارتی دیگر اندکی جزئیات را به تصویر اضافه می‌کند.



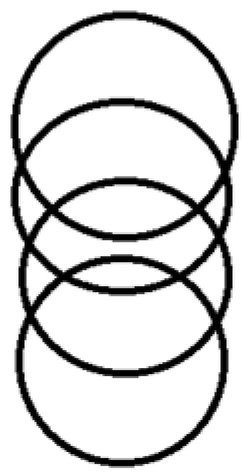

شکل ۴- ۱۱ پویش تصویر ۴-۷ با وضوح های بیشتر، وضوح ۲۰ و ۲۵ نقطه بارنگهای سبز و بنفش به تصویر اضافه شده‌اند.

### ۴-۲-۳ بحث بر روی طول گام‌های مختلف

همانطور که در بخش ۴-۱-۲ عنوان شـــد، گام، فاصـــله‌ی میان دو نقطه‌ی متوالی در مســـیر DSPR اســـت که در هر یک از این نقاط DSPR می‌ایستد و یکی یا هر دو حسگر فراصوت آن شروع به پویش ۱۸۰ درجه کامل می‌کند. در اینجا برای سادگی موضوع فرض می‌کنیم در هر یک از نقاط یاد شـــده، DSPR یک پویش ۳۶۰ درجه انجام می‌دهد. گرچه اصـــل موضـــوع با این فرض تغییر خاصـــی پیدا نمی‌کند. همانند وضـوح، گام نیز یک متغییر قابل تنظیم اسـت. متناسـب با اینکه طول هرگام در مقایسـه با برد حسـگر فراصوت چقدر است سه حالت ممکن، متصور می‌شود: [۱۳]

- برد حسگر فراصوت از طول هر گام بیشتر باشد:

شـکل ۴-۱۲ حالتی را نشـان می‌دهد که برد حسگر فراصوت از طول هر گام بیشتر است. در این شرایط، اگر اطلاعاتی که هر بار پویش توسط DSPR جمع‌آوری می‌شود را به صورت یک دایره فرض کنیم، پویش‌های مکرر DSPR مطابق این شکل خواهد بود. توجه شود که منظور از دوایر، نقاطی هست که DSPR در طی یک پویش، اطلاعاتی از آن‌ها کسب می کند.

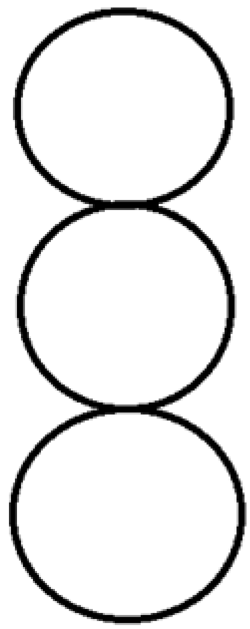


*شکل ۴-۱۲ اطلاعات جمع آوری شده توسط DSPR در حالتی که برد حسگر فراصوت از طول هر قدم بیشتر باشد. هر دایره معرف اطلاعات جمع‌آوری شده در طی یک بار پویش ۳۶۰ درجه‌ی محیط است.*

همانطور که در شـکل ۴-۱۲ ملاحظه می‌شـود، دوایر متقاطع پدید می‌آید. در این حالت بعضـی از نقاط دو و یا بیشـتر از دو بار مورد پویش قرار می‌گیرند(نقاطی که محل تقاطع دوایر هسـتند). این موضـوع سبب می‌شود که امکان اصلاح اطلاعات حاصل از پویش یک نقطه فراهم شـود. به عبارتی دیگر، در این حالت تصـویر دقیق‌تری از محیط اطراف حاصـل می‌شـود و تقریبا تمام نقاطی که در مسیر DSPR قرار دارند پویش می‌شود، در نتیجه تصویر کاملی از محیط فراهم می‌شود. در مقابل، سرعت اجرای عملیات پایین می‌آید. چون DSPR موظف می‌شود در تعداد نقاط بیشتری بایستد و کل محیط را پویش نماید. همچنین به فضای بیشتری برای نگه‌داری داده های حاصل از پویش نیاز است.

- برد حسگر فراصوت با طول هر گام برابر باشد:

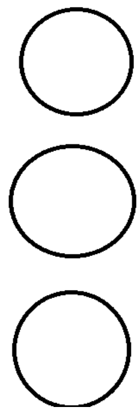
شـکل ۴-۱۳ حالتی را نشـان می‌دهد که برد حسـگر فراصـوت با طول هر گام برابر اسـت. در این حالت دوایر مدل شده با یکدیگر مماس هسـتند. هر نقطه در مسـیر DSPR فقط یک بار مورد پویش قرار می‌گیرد و امکان اصلاح اطلاعات هر نقطه در پویش‌های بعدی وجود ندارد. با این اینکه دقت تصـویر فراهم شده توسط DSPR از محیط نسبت به حالت قبلی پایین‌تر می‌آید، اما هنوز هم تصویر کامل است. سرعت اجرای عملیات نسبت به حالت قبل افزایش می‌یابد. زیرا DSPR باید نقاط به مراتب کمتری را پویش نماید. همچنین به فضای حافظه‌ی کمتری برای ذخیره‌سازی اطلاعات حاصل از پویش نیاز است.



*شکل ۴-۱۳ اطلاعات جمع آوری شده توسط DSPR در حالتی که برد حسگر فراصوت با طول هر قدم برابر باشد. هر دایره معرف اطلاعات جمع‌آوری شده در طی یک بار پویش ۳۶۰ درجه‌ی محیط است.*

- برد حسگر فراصوت از طول هر گام کمتر است:

شکل ۴-۱۴، حالتی را نشان می‌دهد که برد حسگر فراصوت از طول هر گام کمتر است. در این حالت دوایر مدل شده، کاملا مجزا از یکدیگر هسـتند و هیچ فصـل مشـترکی ندارند. دقت تصـویر فراهم شـده از محیط در این حالت از دو حالت قبلی پایین‌تر اسـت زیرا هر نقطه در مسیر DSPR حداکثر یک بار پویش می‌شود. همچنین تصویر فراهم شده کامل نخواهد بود، زیرا در مورد فضای میان دایره‌ها هیچ گونه اطلاعاتی در دست نیست. اما در مقابل، سرعت اجرای عملیات بالاتر و همچنین فضای لازم برای ذخیره سازی به مراتب کمتر از دو حالت قبل است.



*شکل ۴-۱۴ اطلاعات جمع آوری شده توسط DSPR در حالتی که برد حسگر فراصوت از طول هر قدم کمتر باشد. هر دایره معرف اطلاعات جمع‌آوری شده در طی یک بار پویش ۳۶۰ درجه‌ی محیط است.*

در اینجا دومین مصالحه‌ی DSPR مطرح می‌شود. هرچقدر طول گام ها کمتر باشد، تصویر فراهم شده از محیط کاملتر و دقیق‌تر است. در مقابل زمان اجرای عملیات و همچنین میزان فضای حافظه‌ی لازم برای ذخیره سازی داده‌های جمع‌آوری شده از محیط بالاتر می رود. توجه شود که برد عملیاتی حسگر فراصوت SRF08 بین ۳ سانتی‌متر تا ۶ متر قابل تنظیم توسط کاربر است.

## ۴-۳-۴ ارزیابی مکان‌یابی DSPR

در بخش‌های گذشته در مورد اهمیت دقت مکان‌یابی DSPR بسیار بحث شد. در انتخاب موتورهای حرکتی، انکودر و قطب‌نما، مهمترین عامل، دقت عنوان شد. وقتی قرار است DSPR جایگزین عملیات نقشه‌برداری، جستجو و مانند این‌ها باشد، حتما باید بر روی دقت مکان‌یابی آن تاکید ویژه‌ای گردد. در طول انجام این پروژه، به هنگام انتخاب قطعات، همواره بر دقت تاکید شد. حال نوبت به آن رسیده است تا دقت DSPR در مکان‌یابی مورد بررسی قرار گیرد.

منظور از دقت مکان‌یابی، میزان اختلاف میان مختصات واقعی که DSPR در آن قرار دارد و مختصاتی است که DSPR محاسبه می‌کند. به زبان ساده‌تر، این اختلاف، میان مختصاتی است که DSPR در آن قرار دارد و مختصاتی که DSPR گزارش می‌کند و گمان می کند در آن قرار دارد. برای سنجش این دقت از معادله ۴-۱ استفاده شد:

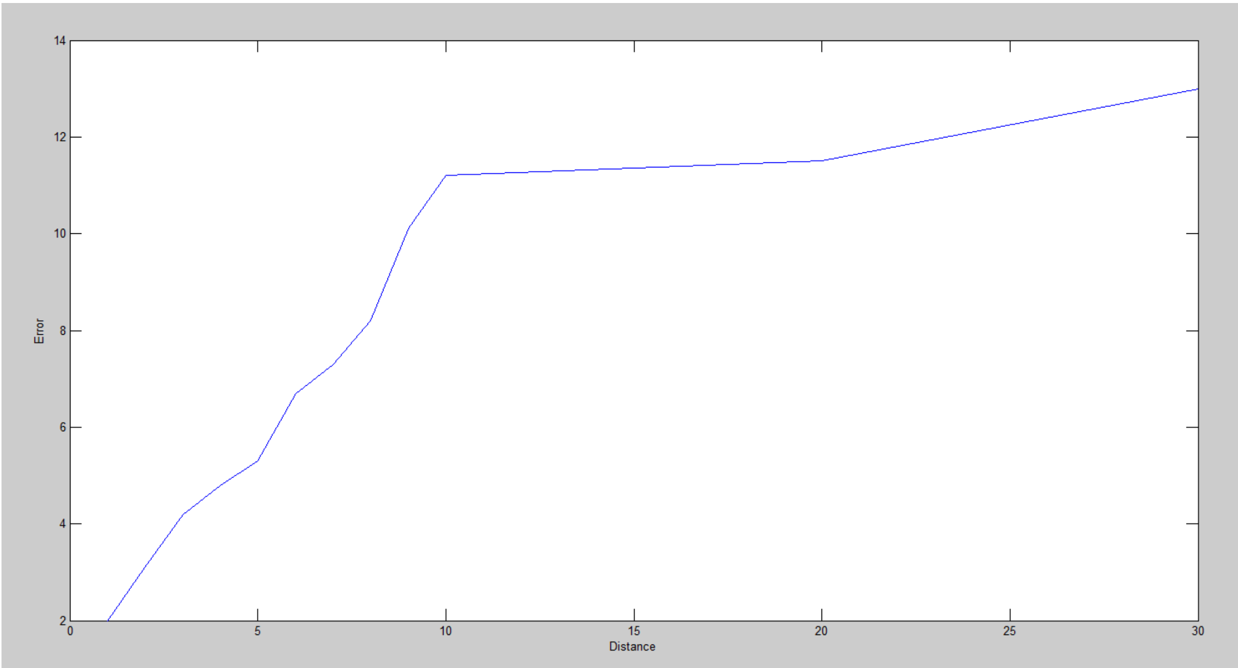
$$e = \sqrt{(X-x)^2 + (Y-y)^2}$$ معادله ۴-۱

در معادله‌ی شماره‌ی ۴-۱، X طول و Y عرض مختصات واقعی DSPR و x طول و y عرض مختصاتی است که DSPR محاسبه و گزارش می‌کند و گمان می‌کند در آن قرار دارد، است. همچنین e میزان خطای مکان‌یابی DSPR می‌باشد. یکاهای x,y,X,Y باید همجنس باشند، مثلا همگی متر، سانتی‌متر، میلی‌متر یا مانند این‌ها باشند. یکای e همجنس سایر یکاهای معادله‌ی ۴-۱ خواهد بود. برای مثال، تصور شود که DSPR در مختصات (100,200) سانتی‌متر قرار دارد. اما انکودرها و سیستم مکان‌یابی آن مختصات (99, 203) را محاسبه و گزارش می‌کنند و گمان می‌برند که DSPR در این مختصات واقع‌شده است. در این حالت طبق معادله شماره ۴-۱ خطای مکان یابی DSPR به صورت زیر قابل محاسبه است:

$$e = \sqrt{(100-99)^2 + (200-203)^2} = 3.16\ cm$$

رابطه‌ی بالا بیان می کند که خطای مکان‌یابی DSPR در گزارش این مختصات ۳,۱۶ سانتی‌متر می‌باشد، البته بدیهی است که خطای DSPR برای گزارش نقاط مختلف و یا حتی گزارش یک نقطه در دفعات مختلف متفاوت است. به همین دلیل از میانگین خطای مکان‌یابی برای

گزارش خطا اسـتفاده می‌شود. شیوه‌ی ارزیابی اولیه‌ی خطای DSPR به این صورت بود که با در نظرگرفتن گام‌های ۲۵ سانتی‌متری، مختصات محاسـبه شـده در محل هرگام توسـط DSPR گزارش می‌شـد و با مختصـات واقعی محاسـبه می‌گردید و خطای مکان‌یابی مورد محاسبه قرار می‌گرفت. سـرانجام ازخطاها میانگین گرفته شـد. این آزمایش برای مسـافت‌های مختلف به دفعات تکرار گردید. میانگین خطاهای مکان‌یابی بدست آمد. شکل ۴-۱۵ خطای مکان‌یابی DSPR را بعد از طی فواصل مختلف نشان می‌دهد.




*شکل ۴ - ۱۵نمودار خطای مکان‌یابی DSPR پس از طی فواصل مختلف، محور افقی فواصل و محور عمودی خطای مکان‌یابی را نشان می‌دهند*

در این شـکل، محور افقی فواصـل مختلف پیموده شـده توسـط DSPR بر حسـب متر و محور عمودی خطای مکان‌یابی DSPR را بر حسب سانتی‌متر به ازای آن فواصل بیان می‌کند. نمودار با افزایش مسافت پیموده شده سیر صعودی به خود می‌گیرد. برای مثال متوسط خطای مکان‌یابی DSPR بعد از پیمایش ۱۰ متر حدودا ۱۱,۳ سـانتی‌متر اسـت. شاید این خطا برای پیمایش ۱۰ متر کمی زیاد باشد، اما نکته‌ی قابل توجه آن اسـت که بعد از این مسـافت، دقت مکان‌یابی تقریبا کاهش چندانی نمی‌یابد و همانطور که ملاحظه می‌شـود پس از پیمایش ۳۰ متر خطای مکان‌یابی به ۱۳ سـانتی‌متر می‌رسـد که خطای نسـبتا قابل قبولی می‌باشـد. بنابراین عمده خطای مکان‌یابی در شـروع عملیات اتفاق می‌افتد.

توجه شود که برای هر یک از فواصل موجود در نمودار شکل ۴-۱۵، ۵ بار آزمایش محاسبه‌ی خطای مکان‌یابی انجام شده است و آنچه در نمودار است خطای میانگین این ۵ بار آزمایش می‌باشد.

# فصل پنجم

# جمع‌بندی پروژه

## ۵-۱خلاصه‌ای از مزایای DSPR

- صـرفه‌جویی در هزینه‌ی سـاخت: هزینه‌ی کل پروژه‌ی ساخت DSPR که جزئیات آن در فصل سوم مطرح شد، به رقم تقریبی ۴۷۰ دلار آمریکا می‌رسد. این رقم در زمان نگارش این مقاله معادل ۱۴۵۷۰۰۰۰ ریال ایران می‌باشد. در مقایسه با روبات‌هایی که DSPR با هدف جایگزین مناسـب برای آنها طراحی شـده اسـت، این رقم، حداقل کمتر از نیمی از میانگین هزینه‌های سـاخت آن روبات ها می‌باشد.

- انعطاف‌پذیری و طراحی ماژولار: در طراحی و پیاده سازی مکانیکی DSPR، از بیش از یک صد قطعه پیچ و مهره استفاده شد. تمامی قطعات مکانیکی با قابلیت نصـب و حذف آسـان طراحی شدند. امکان تغییر مکانیکی DSPR مطابق با شرایط عملیاتی محیط وجود دارد. مکانیک DSPR حتی از یک قطره چسـب تشـکیل نشده است. در طراحی الکترونیکی و برنامه نویسی DSPR، ماژولار بودن به تمام و کمال رعایت گردید. این مسائل، ارتقاء و ایجاد تغییر در مکانیک، سخت‌افزار و نرم‌افزار DSPR را با صرف حداقل هزینه ممکن می‌سازند.

- سـکو بودن: DSPR یک سـکو یا Platform می باشـد. تقریبا تمامی شـرایط و معیارهای یک سـکو در آن به کار رفته اسـت. پیاده‌سازی‌های نرم‌افزاری سطح پایین و تمامی پیاده‌سازی‌های سخت افزاری از دید کاربر محو شده است. کاربر تنها از طریق ارتباط سـریال و با استفاده از جدول دستورات کنترلی با DSPR در ارتباط است. با استفاده از این امکانات کاربر می تواند با هر سیستمی با DSPR ارتباط برقرار نماید و به برنامه‌ریزی آن و اجرای الگوریتم موردنظر خود بپردازد. بدون اینکه حتی اندکی وارد جزئیات شود.

- ابعاد فیزیکی مناسـب: طراحی مکانیکی DSPR مطابق با شـرایط عملیاتی محیط‌هایی است که در آن به انجام وظیفه می‌پردازد. در حال حاضـر این روبات می‌تواند جایگزین روبات‌هایی نظیر نقشـه بردار، جاروب کننده، جسـتجوگر و مانند این‌ها شود. ابعاد فیزیکی DSPR دقیقا مطابق با این شرایط طراحی شده است. به طوریکه بدون کوچکترین تغییر در ابعاد بتواند به کاوش در محیط عملیاتی خود بپردازد.

- حداقل تعداد حسـگر: در سـاخت DSPR تنها از دو عدد حسـگر فراصوت SRF08 استفاده شده است. DSPR می‌تواند تنها با این دو عدد حسگر تصویر اجمالی کامل و نسبتا دقیقی از محیط خود برای پردازش الگوریتم مورد نظر فراهم نماید. **این در واقع مهمترین و برترین مزیت DSPR نسبت به روبات‌های مشابه و جایگزین است.**

- بومی بودن فناوری: تمامی قطعات و فناوری‌های استفاده شده در DSPR به سادگی در بازار بومی ایران قابل تهیه هستند.

## ۲-۵ خلاصه‌ای از معایب DSPR

- سرعت پایین پردازش سخت‌افزار: با وجود اینکه معیار سرعت بالای پردازش سخت‌افزار یکی از معیارهای مطرح شده در طراحی DSPR بود، اما در هنگام پیاده‌سازی به خوبی رعایت نشد. دلیل عمده‌ی این موضوع این بود که برخلاف پیش بینی‌های انجام شده، پیاده‌سازی پردازش توزیع شده نتایج مطلوب را برای افزایش سرعت سخت افزار در پی‌نداشت. تاخیر مالتی‌پلکسر به عنوان شاهراه تمامی مسیرهای بحرانی مدار، سوئیچینگ‌های پی‌درپی میکروکنتلر مرکزی بر روی میکروهای مختلف به هنگام انجام عملیات، تاخیرهای مسیرهای طولانی مدار، ظرفیت‌های خازنی بیش از حد پیش بینی شده‌ی بعضی مسیرها، تاخیر حسگرها، تاخیرهای اجباری برای پایداری سرو موتور به هنگام پایان حرکت، پر شدن بافرهای ارتباط سریال، عدم استفاده از حداکثر فرکانس میکروها به دلیل ایجاد خطا برای حسگرهای فراصوت و قطب نما و عواملی از این دست، باعث کاهش سرعت پردازش سخت‌افزاری مدار شدند. به طوری که یک پویش ۱۸۰ درجه‌ای حسگر فراصوت با حداقل وضوح ۵ نقطه زمان تقریبی ۲۰ ثانیه طول می‌کشد.

- نویز پذیری کانال: کانال ارتباط سریال روبات نویزپذیری نسبتا بالایی را دارد. این امر موجب تغییر بعضی کاراکترهای کنترلی ارسالی می شد. عامل پردازش‌گر الگوریتم با دریافت کاراکترهایی خارج از استانداردهای مشخص شده دچار استثناء (exception) می‌گردید و برنامه متوقف می‌شد. البته در Demo نوشته شده، برای این استثناء یک روال گرداننده‌ی استثنا نوشته شد تا در هنگام بروز این مشکل و دریافت کاراکترهای غیر استاندارد، به تعداد دفعات متناهی تا زمان عدم تکرار استثناء، آخرین فرمان یا درخواست را مجددا ارسال نماید. این امر علاوه بر اینکه تاخیر پردازش روبات را زیاد می‌کرد، در بعضی موارد به دلیل تکرار شدن دفعات استثناء به تعداد زیاد، موجب عدم کارایی روال گرداننده‌ی استثناء و توقف کامل برنامه می‌گردید.

- آسیب‌پذیری فیزیکی انکودرها: در شکل ۳-۳ پیاده سازی مکانیکی انکودرهای روبات مشخص شد. همانطور که در این شکل ملاحظه می‌شود، برای انکودرهای روبات محفظه‌ی خاصی تعبیه نشده بود. این موضوع سبب برخورد پره‌های پلاستیکی انکودرهای روبات با موانع و اجسام کوچک روی زمین می‌شد. در اثر این برخوردها مرتبا پره‌های انکودر روبات آسیب می‌دید و می‌شکست. این موضوع باعث کاهش دقت شمارنده نوری در شمارش پره‌های انکودرها و حتی گاهی از کار افتادن انکودرهای روبات می‌شد.

- عدم محاسبه‌ی نقطه‌ی جرم روبات: به دلیل توزیع ناهمگن بار در روبات و عدم محاسبه‌ی نقطه‌ی جرم آن، انحراف روبات از راستای اصلی به مراتب صورت می‌گرفت. این امر موجب فعال شدن سیستم اصلاح حرکت و متوقف‌شدن الگوریتم اصلی و در نهایت تاخیر در اجرای عملیات روبات می‌شد.

## ۵-۳ خلاصه‌ای از کاربردهای DSPR

DSPR می‌تواند به عنوان یک سـکو جایگزین روبات‌های مشـابه برای اجرای عملیات نقشـه‌برداری، جاروب‌کردن، جستجو و مانند آنها باشـد. برای مثال با نسـخه‌ی قبلی این سکو که MSPR نام داشت یک ماز عملیاتی حل شد. ماز عملیاتی یک ماز واقعی با تمام محدودیت‌های ساخت می‌باشد، از جمله اینکه فاصله‌ی میان دیواره‌ها در آن یکسان نیست، ارتفاع تمام دیواره‌ها با هم برابر نیستند، راه‌ها الزاما خطوط شکسته نیستند و می‌توانند خطوط منحنی باشند و مواردی از این قبیل.[۱۳]

می‌توان از این روبات برای مقاصـد تحقیقاتی، آزمایش الگوریتم‌های جدید، سـنجش کارایی الگوریتم‌ها و موارد مشابه آن استفاده کرد. همچنین این روبات می‌تواند در مقاصد آموزشی روباتیک و طراحی الگوریتم استفاده شود. چون سطح دسترسی کاربر به سخت افزار بسته شده اسـت، بنابراین نوآموزان با پیچیدگی‌های سـخت‌افزاری روبرو نخواهند شـد و می‌توانند به راحتی الگوریتم‌های خود را از کامپیوتر شخصی اجرا نمایند و نتیجه را در DSPR مشاهده کنند.

## ۵-۴ اهداف اجرایی آینده برای این پروژه

در زمان نگارش این پایان نامه، طراحی نسـل دوم این روبات سکو با نام DSPR 2.0 آغاز شده است. DSPR 2.0 از لحاظ ساختار تا حد بسـیار زیادی شـبیه به DSPR اسـت. در DSPR 2.0 سـعی شده است تمامی معایب DSPR برطرف شود. از جمله این راهکارها محاسبه مرکز جرم برای جلوگیری از انحراف روبات و حفظ تعادل آن است. همچنین با راه اندازی تمامی قطعات و ماژول‌ها در فرکانس ۱۶ مگاهرتز و استفاده از میکروکنتلر قدرتمند ATMega 128 و پردازش متمرکز به جای استفاده از چندین میکروکنتلر با پردازش توزیع شده (کاری که در این پروژه انجام شـد) سـعی در افزایش سـرعت پردازش سـخت افزار می‌باشـد. ماژولار بودن پروژه با هدف افزایش سرعت پردازش سخت افزار، از سطح سخت‌افزار به سطح نرم‌افزار منتقل می‌شود. مطالعات در مورد نحوه‌ی بهینه‌ی کد کردن اطلاعات برای حداقل کردن تاثیر نویز در جریان است. موتورهای حرکتی روبات با موتورهایی با گشـــتاور بالاتر تعویض خواهند شـــد. چرخ‌های روبات با چرخ‌هایی مشـــابه اما با قطر بزرگتر جایگزین خواهند شد تا سرعت روبات افزایش یابد. برای پره‌های انکودر محفظه‌ای امن طراحی خواهد شد.

سـعی بر این اسـت تا DSPR 2.0 در محیط‌های شـبیه‌سـازی مطرح مانند webot مدل‌سـازی شـود. همچنین نرم‌افزار شـبیه‌سـازی جداگانه‌ای برای آن در محیط OpenGl در دست طراحی است.

در پروژه‌ی DSPR 2.0 برای اولین بار از سیستم چندعامله استفاده خواهد شد. در ابتدای کار هدف بر این است تا دو عدد DSPR 2.0 از طریق یک کانال طراحی شـده ارتباط بی‌سیم، با یکدیگر در ارتباط باشند. ماموریت‌ها به صورت مشترک انجام شود و محیط عملیاتی تقسیم بندی گردد. روبات‌ها کدگذاری شوند و هر روبات دارای شناسه‌ای مشخص نزد روبات دیگر و سرور باشد.

# فهرست منابع


.۱ RoboCup-Technical-Committee, *RoboCup Standard Platform League (Nao) Rule Book*. (2011 rules, as of May 31, 2011), Robocup Organization: Istanbul, Turkey.

.۲ PublicRelationDivision, *ASIMO Technical Information*. 2007, Honda Motor Co.Ltd.

.۳ ALDEBARAN, *Datasheet NAO humanoid*. 2005, ALDEBARAN Robotics

.٤ TheTechnicalCommittee, *RoboCup Logistics League sponsored by Festo*. 2013, Netherland: Robocup Organization.

.٥ MixedRealityOrganization, *Mixed Reality Competition Rules*. 2010, Singapoor Robocup Oraganization.

.٦ PARALLAX, *Eddie Control Board with Power Connector (#28993)*, PARALLAX, Editor. 2012.

.٧ PARALLAX, *Eddie Robot Platform*, PARALLAX, Editor. 2013: USA.

.٨ PARALLAX, *Parallax Service for RDS 4 Installation Guide*, PARALLAX, Editor. 2013: USA.

.٩ Yahya Hassanzadeh-Nazarabadi, Sajjad Gharibnavaz, and A. Alizadeh. *Dual Sensor Platform Robot Technical Document Paper*. in *8th Robocup IranOpen*. 2013. Tehran,Iran.

.۱۰ Yahya Hassanzadeh-Nazarabadi, Ahmad Charehsaz-Avari-Firoozeh, and A. Mafi, *Mono Sensor Platform Robot*, in *The 16th IEEE Mediterranean Electrotechnichal Conference*. 2012: Yasmin-Hammamet, Tunisia.

.۱۱ MicrochipTechnology, *8-bit AVR Microcontroller with 32K Bytes In-System Programmable Flash*, ATMEL, Editor. 2004: USA.

.۱۲ Hassanzadeh-Nazarabadi, Y ,.*Parse A Technichal Document Paper*. 2009, 4th Robocup IranOpen: Qazvin, Iran.

.۱۳ Hassanzadeh-Nazarabadi;, Y., et al., *Solving operational maze using mono sensor platform robot*, in *4th international conference on computational intelligence, communication systems and networks*. 2012: Phuket, Thailand.

.۱٤ Fitzgerald, C.k., Stephen Umans, *Electric Machinery* 6 ed. 2005: Hardcover.

۱٥. مهدی‌زاده, م., روباتیک, اصول طراحی و مبانی ساخت. ۱۳۸٦: پاژ.

.۱٦ Hassanzadeh-Nazarabadi, Y., *Parse A Technichal Document Paper*٥ ,۲۰۱۰ .th Robocup IranOpen: Tehran, Iran.

۱۷. حسن‌زاده‌نظرآبادی, ی., et al., ارائه ی روش بهینه ی نقشه برداری در روبات مین یاب هوشمند, in پانزدهمین کنفرانس دانشجویی مهندسی برق ایران. ۱۳۹۱: کاشان, ایران.

.۱۸ David.A.Hodges, H.G.J., Resve.A.Saleh, *Analysis and Design of Digital Integrated Circuits*. 3rd ed. 2005: McGraw Hill.

۱۹. مالکی, ع., آموزش *Protel DXP*, ed. چ. دوم. ۱۳۸٦, تهران، ایران: کانون نشر علوم.

.۲۰ Semiconductors, V., *Reflective Optical Sensor with Transistor Output*, in *TCRT5000(L)*, V. Semiconductors, Editor. 2000: Germany.

.۲۱ Sharp, *GP2S04 GP2S06 GP2S07 GP2S09 GP2S10*, Sharp, Editor. 1998: USA.

.۲۲ Panasonic, *CNZ2179 (ON2179) Reflective Photosensor*, Panasonic, Editor. 2001: USA.

۲۳. ثنایی‌نژاد, م.ا.ف., فاصله یابی با دقت و برد مطلوب به کمک حسگر مادون قرمز, in دوازدهمین کنفرانس دانشجویی مهندسی برق ایران. ۲۰۰۹, سازمان علمی دانشجویی مهندسی برق ایران: تبریز، ایران.

۲٤. سعیدی.ف، ع.ا.م.د.م., ریاضی عمومی. چهارم ed. 1387, مشهد، ایران: موسسه چاپ و انتشارات دانشگاه فردوسی مشهد.

.۲٥ SHARP, *GP2Y0A02YK*, SHARP ,Editor. 2006: USA.

.۲٦ Brey, B., *The Intel Microprocessors: Architechture Programming and Interfacing*. 7 ed. 2006, USA: PEARSON.

.۲۷ DevantechLtd, *SRF08 Range Finder*, DevantechLtd, Editor. 2005: USA.

.۲۸ MicrochipTechnology, *8-bit AVR Microcontroller with 8K Bytes In-System Programmable Flash*, ATMEL, Editor. 2004: USA.

.۲۹ corporation, F.s., *KA78XX/KA78XXA 3-Terminal 1A Positive Voltage Regulator*, F. semiconductor, Editor. 2001: USA.

۳۰. ره‌افروز, ا., میکروکنتلرهای *AVR* 1386, تهران، ایران: نص.

.۳۱ STMicroelectronics, *L298n DUAL FULL-BRIDGE DRIVER*, STMicroelectronics, Editor. 2000: USA.

.۳۲ W.H.Hayt, J.E.K., S.M.DURBIN, *Engineering Circuit Analysis*. 6th ed. 2006, India: McGraw-Hill.

.۳۳ R.L.BOYLESTAD, L.N., *ELECTRONIC DEVICES AND CIRCUIT THEORY*. 10th ed ,۲۰۰۹ .USA: PEARSON.

.۳٤ HOPE-RF, *HM-TR transport wireless datalink module*, H. Microelectronics, Editor. 2002: China.

.۳٥ NXP, *74HC4052; 74HCT4052 Dual 4-channel analog multiplexer/demultiplexer*, NXP, Editor. 2012: USA.

۳٦. و.استالینگ, انتقال داده ها و شبکه های کامپیوتری. هفتم ed. 1387, تهران، ایران: علوم رایانه.

۳۷. آ.اس.تننباوم, شبکه های کامپیوتری. ٤ ed. 1388, تهران، ایران: علوم رایانه
۳۸. حسن‌زاده‌نظرآبادی, ی. and ن. یزدان‌پناه, طراحی الگوریتم روبات مین یاب هوشمند, in اولین کنفرانس ملی کاربردهای مهندسی نرم افزار. ۱۳۸۷: لاهیجان, ایران.
۳۹. ZC-Sensors, *ZCT245AN-232 2-axis Tilt Sensor* S.Z.E.C. Ltd, Editor. 2007: China.
٤٠. AltiumDesigner, *Multi-Sheet Design*. 2012, Altium USA.
٤١. AltiumDesigner, *3D Mechanichal CAD*. 2012, Altium USA.